Predictive Networks: *Networks that **Learn**, **Predict** and **Plan***

JP Vasseur (jpv@cisco.com), PhD - Cisco Fellow AI/ML – Release v3.0 – August 2023

Initial Release Sept 2021

***Abstract**: Since the early days of the Internet, beginning with ARPANET in 1970, Network Recovery/Reliability has been a paramount concern. The Routing Protocol Convergence Time — the time required to detect and reroute traffic to address link/node failures — has improved significantly, decreasing from dozens of seconds to mere milliseconds. Numerous protocols and technologies have been developed and widely deployed with the aim of mitigating the impact of network failures. While these advances have significantly bolstered overall network availability, they predominantly depend on reactive approaches. Here, failures must first be detected, followed by traffic rerouting along an alternate path. This path, whether pre-computed or determined on-the-fly, may lack certain guarantees. Furthermore, it's crucial to broaden the failure scope from merely "path alive" to "paths that result in subpar application experiences" (often termed "Grey failures"). Extensive experiments have revealed that many paths, although deemed "alive" by Layer-3 protocols, underperform, thereby degrading user experiences. Contrasting with traditional approaches that have persisted since the Internet's inception, the innovative "Predictive Networks" paradigm seeks to reroute traffic *prior* to an anticipated failure. It ensures that the alternate paths meet application Service Level Agreement (SLA). This white paper delves into the rise of Predictive Networks, which leverage learning technologies to forecast a vast array of failures. The initial version of this paper was released in June 2021, subsequent to 1.5 years of thorough investigation. This updated version incorporates fresh results and examples stemming from the large-scale deployment of such state-of-the-art technologies. These are now integrated into commercial products as of 2023, advancing well beyond mere experimental stages. The potential for Learning, Predicting, and Planning might herald a paradigmatic shift for future networking. The concluding section explores the potential application of these technologies in emerging networking domains such as SASE, home networking, and even BGP for Inter-AS traffic.*

1 A new world, with new challenges

Network recovery has been a topic of high interest in the networking industry since the early days of the ARPANET in 1970. Nonetheless, the paradigm has not changed much: first, a failure (path disruption) must be detected, followed by traffic rerouting along an alternate path; such a path can be either pre-computed (i.e., protection) or computed on-the-fly (i.e., restoration).

Let us first discuss the concept of “Failure detection”. The most efficient approach is to rely on inter-layer signaling whereby a layer n is able of detecting a failure and propagates to upper payer n+1 (e.g. the physical layer detects an optical failures and propagates to IP layer). Unfortunately, a large proportion of failures impacting a link or a node are not detectable by lower layers. Thus, other techniques such as Keep Alive (KA) messages have been designed. There is clearly no shortage of KA mechanisms implemented by routing protocols such as OSPF, ISIS or BFD.

KA have their own shortcomings related to their parameter settings: aggressive timings introduces a risk of oscillations of traffic between multiple paths upon missing few KA messages, a real challenge on (lossy) links where packet loss is not negligible (such issues can be partially mitigated using some form of hysteresis). Once the failure is detected, a plethora of techniques have been developed such as Fast IGP convergence (OSPF or ISIS using fast LSA/LSP generation, fast triggering of SPF, incremental SPF to mention a few), MPLS Fast Reroute (using a 1-hop backup tunnel for link protection or next-next hop backup tunnels for node protection), IP Fast Reroute (IPRR) or other protection

mechanisms used at lower layers (1+1 protection, 1:N, etc.). Those techniques have proven their efficacy at minimizing downtimes, allowing for fast convergence times in the order of a few (dozens of) milliseconds, while potentially guaranteeing equivalent SLA on alternate paths (e.g., MPLS TE with bandwidth protection).

Unfortunately, there is a critical category of failures that impact application experience (aka Quality of Experience QoE) that remain undetected/undetectable. The notion of grey failures is discussed later in this document. These grey failures (sometimes also referred to as brown failures) may have a high impact on application experience because of high packet loss, delay or jitter ***without*** breaking the link/path connectivity (and thus they are not detected by the aforementioned technologies as "Failure") including transient phenomenon such as MIF (as discussed in). In this case, most (if not all) KA-based approaches would not detect such a failure, leaving the topology unchanged and the traffic highly impacted even though a preferable alternate path may exist.

Existing solutions such as Application Aware Routing or AAR (SD-WAN: Application-Aware Routing Deployment Guide, 2020) rely on the use of network-centric mechanisms such as BFD and HTTP probes to evaluate whether a path meets the SLA requirements of an application using a so-called SLA Template. The most common approach consists in specifying some hard threshold values for various network centric KPIs such as the delay, loss and jitter averaged out over a given period of time (e.g., delay average computed over 1h should not exceed 150ms one-way for voice). Such templates may be highly debatable and the use of statistical moments such as average or other percentile values have the undesirable effect of smoothing out the signal and losing the necessary granularity to detect sporadic issues that do impact the user experience. One may then shorten probe frequencies, use higher percentiles but the granularity is unlikely to suffice for the detection of grey failures impacting the user experience (e.g., a 10s high delay/packet loss would highly impact voice experience while not being noticed when probing every 1-minute or even more frequently but averaging out over 1h periods). Note that new approach consisting in using a ML algorithm to dynamically learn the QoE using cross-layer telemetry sources is discussed in (Vasseur J. , et al., 2023).

Still, AAR is a great step forward when compared with the usual routing paradigm, according to which traffic rerouting occurs only in presence of dark failures (i.e., path connectivity loss). Note that AAR is a misnomer since the true application feedback signal is not taken into account for routing; path selection still relies on static network metrics and SLA templates are a posteriori assessed and verified as explained above. AAR is *reactive* (the issue must first be detected and must last for a given period of time for a rerouting action to take place) with no visibility on the existence of a better alternate path: rerouting is triggered over alternate paths with no guarantees that SLA will be met once traffic is rerouted onto the alternate path. This last component is of the utmost importance. With most reactive systems using restoration such as IP, upon detecting that the current path is invalid (e.g., lack of connectivity, or not meeting the SLA) the secondary path is selected with no guarantee that it will keep satisfying the SLA once traffic will be rerouted.

2 Path computation

How are paths/routes being computed in the Internet? This is the task of routing protocols. IGPs (ISIS or OSPF) make use of a Link State DataBase (LSDB) to compute shortest paths using the well-known Dijkstra algorithm. Link weights/costs statically reflect some link properties (link bandwidth, delay, …). More dynamic solutions using control plane Call Admission Control (CAC) such as MPLS Traffic Engineering allows for steering traffic along Labeled Switched Path (LSP) computed using constraint-shortest paths (that may be computed using a distributed head-end driven Constrained SPF (Shortest Path computation)). Alternatively, such TE LSPs may also be computed using a Path Computation Element (PCE) for Optical/IP-MPLS layers, intra and inter-domain. Constrained shortest paths may even be recomputed according to measured bandwidth usage (e.g., Auto-bandwidth). The PCE is a central engine that gathers topology and resources information to centrally compute Traffic Engineering LSP (TE LSP) paths. The PCE can be stateless or stateful performing global optimization of traffic placement in the network.

Inter-AS routing has been relying on BGP (an Exterior Gateway Protocols) that has been widely deployed to exchange routes between Autonomous Systems (AS) for the past four decades. BGP has been scaling remarkably well and as of 2022 routing tables comprise up to 915K IPv4 prefixes and 152K Ipv6 prefixes. Inter-AS paths will involve a set of AS that are each managed by distinct administrative domains using disparate traffic engineering policies, making the optimization of end-to-end path extremely challenging, if not impossible.

3 Before Predicting, the network must be able to Learn

3.1 Learning in the human brain

The human brain is without a doubt the most advanced learning/predicting engine. Despite considerable breakthrough discoveries in Neuroscience over the past few decades, the learning strategies of the human are not compltelety understood. The learning capabilities manifest themselves in several ways:

- The Hebbian theory related to synaptic plasticity has been a key principle in neuroscience *"What fire together wire together"*; thanks to synaptic plasticity, neural networks are formed (wire together) dynamically thus allowing us to *learn* and also unlearn (for example thanks to synaptic downscaling during sleep).
- The brain is made of a collection of networks (e.g., Default mode, sensimotor, visual, attention, …) that are also highly dynamic: new links and connections (synaptic plasticity), new nodes/neurons (neurogenesis in areas such as the hippocampus), and even new networks topologies (rewiring).

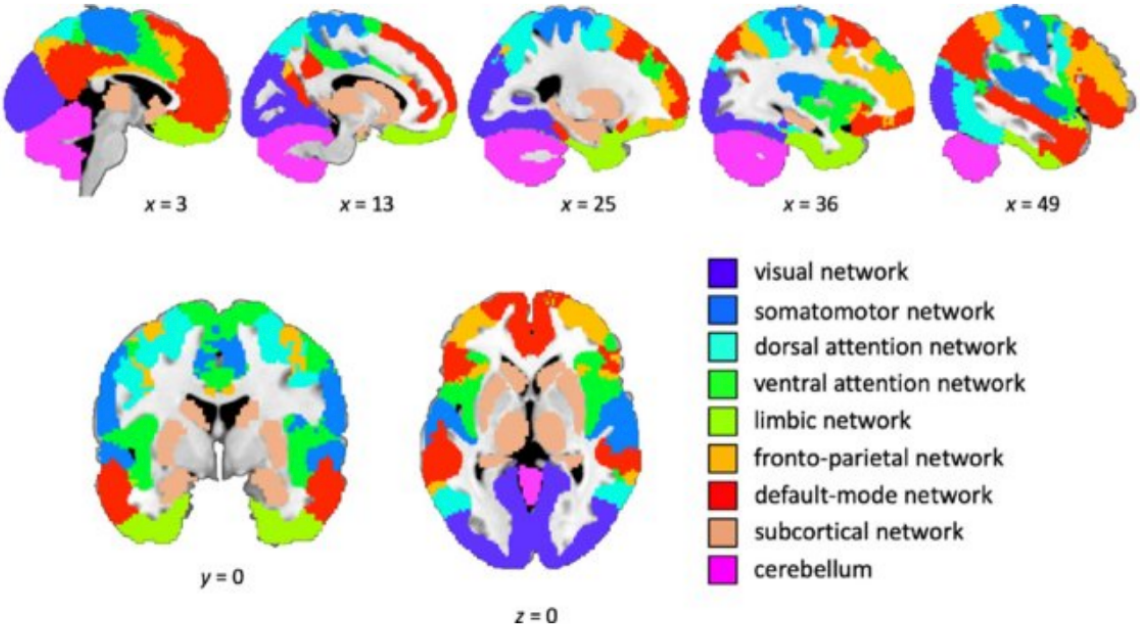

Figure 1: Brain Networks

The human brain is not only an impressive learning engine but also a remarkable predictive engine: the simple action of grabbing an object involves a complex series of predictive actions (e.g., just anticipating the shape of the object), or the prediction of the position of the ball when playing tennis due to the high processing times required by vision.

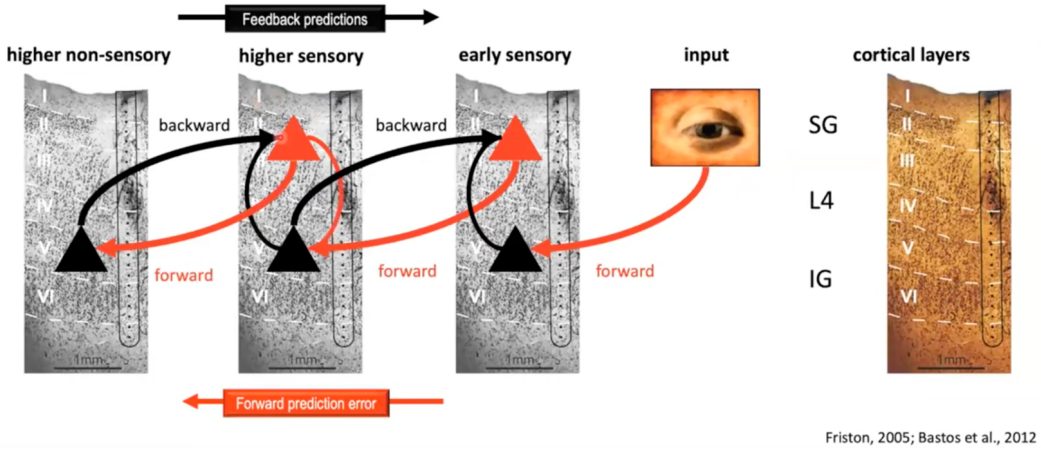

Figure 2: Brain Predictive Coding across cortical layers

A well-known theory named “Predictive Coding” and depicted in Figure 2 shows that some areas of the brain get inputs from other regions, provide predictions that get adjusted according to sensing. A number of theories on the critical topic of “Predictive Coding” such as the Free Energy Principle have been first explored using a Bayesian approach by Friston in 2005. Other theories state that the ability to learn with very few labels relies on the brain’s ability to build a predictive model of the world mostly thanks to observation allowing a human to quickly learn and predict. This is in contrast with existing supervised machine learning techniques in Artificial Intelligence (AI) that require a massive amount of labeled data to train a model. Similarities with a predictive Internet will be shown later in the document.

The same challenge exists for reinforcement learning techniques that also require huge amount of action/feedback to efficiently train a model. Such challenges gave rise to new approaches in Machine Learning such as Self-Supervised learning where knowledge/learning is performed with no supervision on large amount of data that are then augmented with additional labels.

Other forms of predictions involving hierarchical structures (Hierarchical coding) with interaction between sensing and prediction (in different brain areas communicating via different layers of the neocortex) are also very well-known in vision, auditory pathways and Natural language processing.

Higher level predictions are also known to be performed in the PreFrontal Cortext (PFC). Without elaborating too much, the ability to decide (Automate in networking terms) is also a key ability human (and animals) have that heavily involves the Orbito Frontal Cortex (OFC).

The interplay between AI and Neuroscience, which promises significant benefits for each discipline, is discussed in detail in (Vasseur J. , AI and Neuroscience: the (incredible) promise of tomorrow, 2023).

3.2 The much-needed ability for networks to learn

It is an unescapable fact that most of the control plane networking technologies do not incorporate learning. Instead, today’s technologies and protocols rather focus on the ability to **react** as quickly as possible using current states that do not leverage any form of learning and predictive capabilities.

Imagine a human brain incapable of learning and rather just reacting!

The Internet and networks in general have not been designed with the ability to learn (from historical data), use models (of the network) and predict. Data collection has most often been used for monitoring and troubleshooting of past events. Protocols simply do not learn. There are some minor exceptions such as packet retransmissions (backoff) at lower transport layers, the ability to estimate/learn the instantaneous bandwidth along a given path, the use of hysteresis or some protocols trying to estimate capacities before using a path such as (Cardwell, Cheng, Gunn, Yeganeh, & Jacobson, 2016). But for the most part, protocols have adaptive/reactive behaviors according to a very recent past, without learning/modelling capabilities.

Networks are highly ***diverse*** and ***dynamic***: in a multi-cloud highly virtualized world where the network keeps changing and applications constantly move, it has never been so important to equip the Internet with the ability to learn. Indeed, because networks are highly diverse and dynamic the necessity to learn and adapt using technologies such as Machine Learning (ML) is of the utmost importance. Similar properties were observed in other networking areas: for instance, the ability to detect abnormal joining times or percentage of failed roaming in Wifi networks cannot rely on the use of static thresholds. In 2018 ML-based tools such as Cisco AI Network Analytics (Cisco AI Network Analytics, n.d.)) were developed where ML models were used to learn the expected values of networking performance variables according to a number of parameters reflecting the network characteristics. The ability to learn allowed for a dramatic reduction of unnecessary noise (false alarms due to the use of inappropriate static thresholds). Although the use cases for applying ML to the WAN are different (predict issues), this highlights the need to learn and dynamically adapt to the network characteristics.

4 Predictive Networks

Starting with a famous quote attributed to Niels Bohr: *“It is hard to make predictions, especially about the future”*. Cisco has invested considerable research and development resources since 2019 to investigate the ability for networks to “Learn, Predict and Plan”.

The ability to predict requires building models trained with large amount of data. First, an unprecedented analysis has been made on millions of paths across the Internet, using different networking technologies (MPLS, Internet), access

types (DSL, fiber, satellite, 4G), in various regions and thousands of Service Provider networks across the world. The objective was first to evaluate the characteristics of a vast number of paths (Dasgupta, Kolar, & Vasseur, Jan 2022). (Vasseur & Vasseur, 2023) provides an overview of such analysis. Figure 3 shows a few approaches for data path statistical models. More details can be found in the internet dynamics study (Vasseur & Vasseur, 2023; Vasseur & Vasseur, 2023; Vasseur & Vasseur, 2023; Vasseur & Vasseur, 2023).

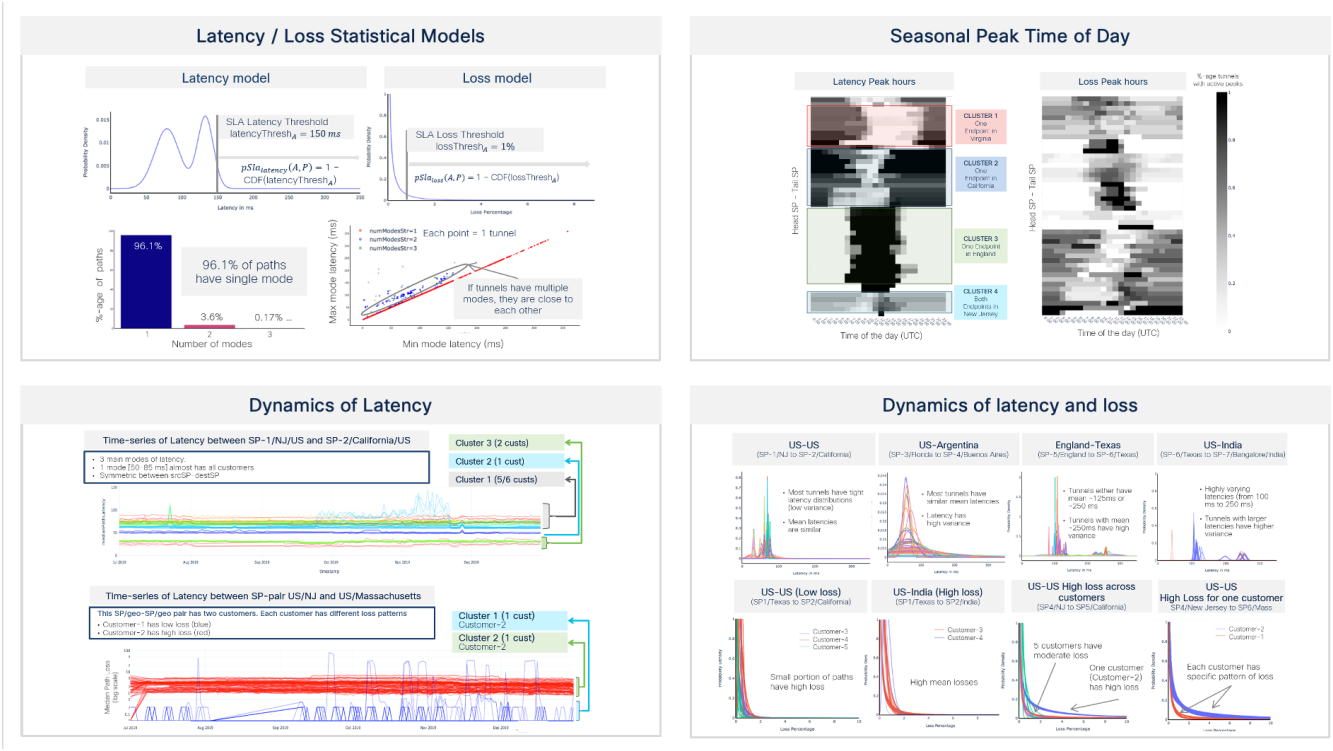

Figure 3: Models of Path dynamics in the Internet

More advanced models relying on a variety of features and statistical variables have been designed (e.g., spectral entropy, welch spectral density, …) along with their impact on application experience (Dasgupta, Kolar, & Vasseur, Jan 2022).

The fact that path "quality" across the Internet is diverse and varies over time is not new. But the main key take-away lies in the ability to quantify across space and time using a broad range of statistical and mathematical analysis.

Simply said, predicting/forecasting refers to the ability to make assumptions on the occurrence of events of interest (e.g., dark/grey failures) using statistical or ML models learned using historical data.

4.1 Short-term versus Mid/Long-term predictions

The forecasting horizon (how much time in advance should the prediction be made) is one of the most decisive criteria. Forecasting with very low time granularity (e.g., months) is usually less challenging especially when the objective is to capture trends. On the other side of the spectrum, a system capable of forecasting a specific event (e.g., failure) impacting the user experience is far more interesting (and challenging).

A number of approaches have been studied since 2019 both for short- and long-term predictions in predictive networks using both statistical and ML models.

For example, ML algorithms such as Recurrent Neural Network (RNN) like Long Short-Term memory (LSTM) may be used for forecasting, using local, per-cluster or even global models (Figure 4 shows an LSTM with Auto-encoder model used for prediction). The LSTM was trained using a large number of features such as various statistical moments (mean, median, percentiles) for KPI such as delay, loss and jitter computed over short and long-term periods of time, per-interface features, temporal features, categorical features related to Service Providers or even traffic-related features.

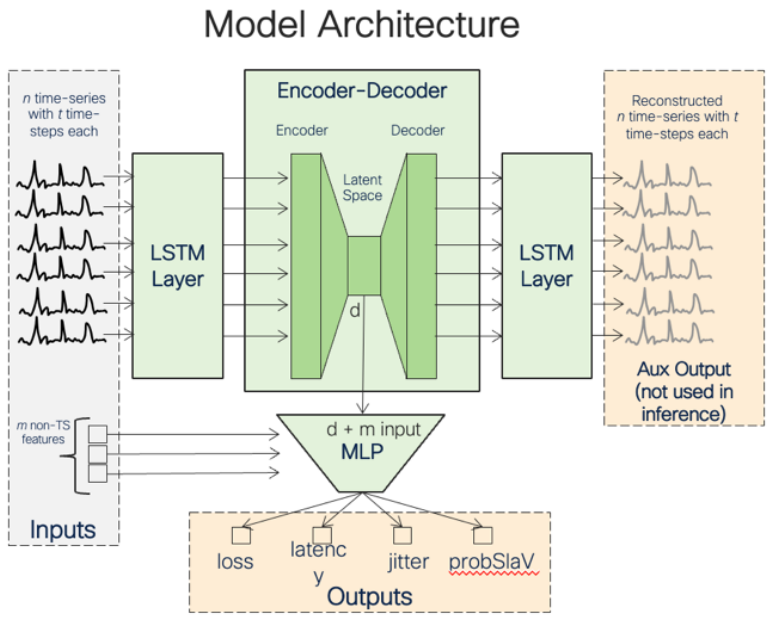

Figure 4: LSTM with Auto-encoder algorithm used for prediction

We have also developed other approaches such as State Transition Learning for forecasting failure events by observing the prominent subsequence of network state trajectories that may lead to event of interest (e.g., failures).

Mid- and Long-term prediction approaches ought to be considered whereby the system models the network to determine where/when actions should be taken to adapt routing policies and configuration changes in the network in light of the observed performance and state (i.e., Internet behavior). Such predictions then allow for making recommendations (e.g., change of configuration or routing policies) that will improve the overall network Service Level Objectives (SLO) and application experience.

Mid- and Long-term predictions (days/weeks) have proven to be highly beneficial, although less efficient than short term recommendations (hours) that deal with short term predictions and remediations. Such systems must consider a series of risk factors including the stability of the network and traffic pattern in order to minimize the risk of predictions that would be quickly outdated. This contrasts with a short-term predictive engine used for “quick fixes” and avoidance of temporary failures; such an approach allows for covering a number of failure scenarios with early signs that would be enabled with full automation. Indeed, forecasting a mid-long term issue (in days/weeks) allows for manual intervention from the network administrator. In contrast, planning for issues forecasted a few minutes/hours requires the enablement of a close loop (full automation) network administrators may not (yet) always want to embrace.

4.2 Forecasting accuracy

Forecasting accuracy has been a recuring topic. Any forecasting system will make forecasting errors. However, predictive engines should be designed so as to make trade-offs between True Positives (TP), False Positive (FP) and False Negative (FN).

TP occurs when the model predicts a failure, and a failure occurs. FP means that a failure has been predicted and does not occur whereas a FN refers to the opposite situation (a failure happens that has not been predicted). For example, a Machine Learning (ML) classification algorithm may be tuned to deal with the well-known Precision/Recall tension where Precision=TP/(TP+FP) and Recall=TP/(TP+FN). In other words, the algorithm must be tuned to favor either Precision or Recall. Cisco’s predictive engine favors Precision over Recall, a safe approach for highly minimizing the risk of FPs: the system has been optimized so as to not always predict all failures but having the highly desirable objective of having almost no false positives: when a failure is predicted, it almost always happens. *What does that mean for recall?* As expected, such a system cannot predict all issues but with existing reactive approaches there is no prediction at all! Said differently, a safe approach consists in building predictive engines that will keep increasing their Recall while not doing any compromise on Precision.

In a live system, such as the one Cisco has developed, other criteria must be taken into account and time granularity is of the utmost importance both for telemetry gathering and time to react (triggering close loop control) with tight implications on the architecture.

4.3 Are all failures predictable?

From a **pure theoretical standpoint**, yes, since true randomness is extremely rare in nature (in quantum physics true randomness is well-known). So, events such as failures are usually caused by other events that may theoretically be detectable. The networking day-to-day reality is, of course, ***very*** different. In most cases, events indicative of some upcoming failures may exist, but they are not always monitored. Moreover, the forecasting horizon is not always compatible with the ability to trigger some actions; even a fiber cut may be predicted by monitoring the signal in real-time but a few nanoseconds before the damage, not leaving enough time to trigger any recovery action, even if a predictive signal exists. In reality, the ability to predict events is driven by the following factors:

- Finding "Signal" in telemetry (when such telemetry exists) with high SNR,
- Computing a reliable ML model with good Precision/Recall,
- Designing an architecture at scale supporting a Predictive approach (this last aspect should not be overlooked; there is a considerable gap between an experiment in a lab and the scale of the Internet).

Predicting consists in finding early signals with predictive power used to build a model and producing a given outcome (e.g., component X will fail within x ms, or probability of failing of component Y is Pb) using classification and/or regression approaches. The ability to predict raises a number of challenges Cisco managed to overcome thanks to a decade of deep expertise in Machine Learning and analytics platforms.

4.4 Could a proactive action potentially negatively impact other paths?

Many other dimensions must be considered when designing a predictive system. For example, would the proactive action of rerouting traffic from a path A to a path B potentially save the traffic from path B from poor QoE but also impact the traffic already in place along path B? This is one of the constraints the predictive algorithm must take into account. Such congestion (due to the sum of the traffic) may take place locally (shared interface), along the last mile to the Service Provider (SP) PoP or even downstream in the network/Internet (although less likely). A predictive may then not just predict poor QoE along a given path but it must also predict significantly better QoE (or lower probability of poor QoE) on alternate paths taking into account under new traffic conditions. Several algorithmic approaches may be envisioned as a computing model for a target taking into account the traffic conditions (and in particular) the volume of traffic, or even reinforcement learning where the system learns rewards/penalties according to the observed condition after a pro-active reroute.

A very common question is whether proactively rerouting traffic onto a set of alternate paths may lead to moving congestion points in the network. In theory, such a situation could happen when adopting a very naive approach for triggering rerouting. In reality, networks are highly granular made of very large number of paths, which of course, share resources. The probability though to shift congestion because of the proactive rerouting of a set of paths at the edge of the network (in the case of the SD-WAN use case) is highly unlikely and the sensitivity of path to volume of traffic in such conditions has been shown to be very small in reality. Still because the decision to trigger pro-active actions is made by a central engine, it is straightforward to avoid congestion appearing by simply distributing the load over available resources.

4.5 Path computation with Predictive Networks in SD-WAN

The concept of Predictive Networks is not limited to a specific "use case" and could be applied to several contexts, use cases and networks using a broad range of technologies. In this paper revision, we focus on Predictive Networks technologies for SD-WAN as shown in Figure 5.

In this classic example, a remote site (called "Edge") is connected to a Hub via SD-WAN using several tunnels each having various properties. It is also quite common for the edge to be connected to the public Internet via an interface sometimes called DIA (Direct Internet Access). In such a situation, the traffic destined to, for example, a SaaS application S can be sent along one of the tunnels from the edge to the Hub (backhauling) or via the Internet using IP routing, or even using a (GRE or IPSec) tunnel to a Security Cloud (SASE).

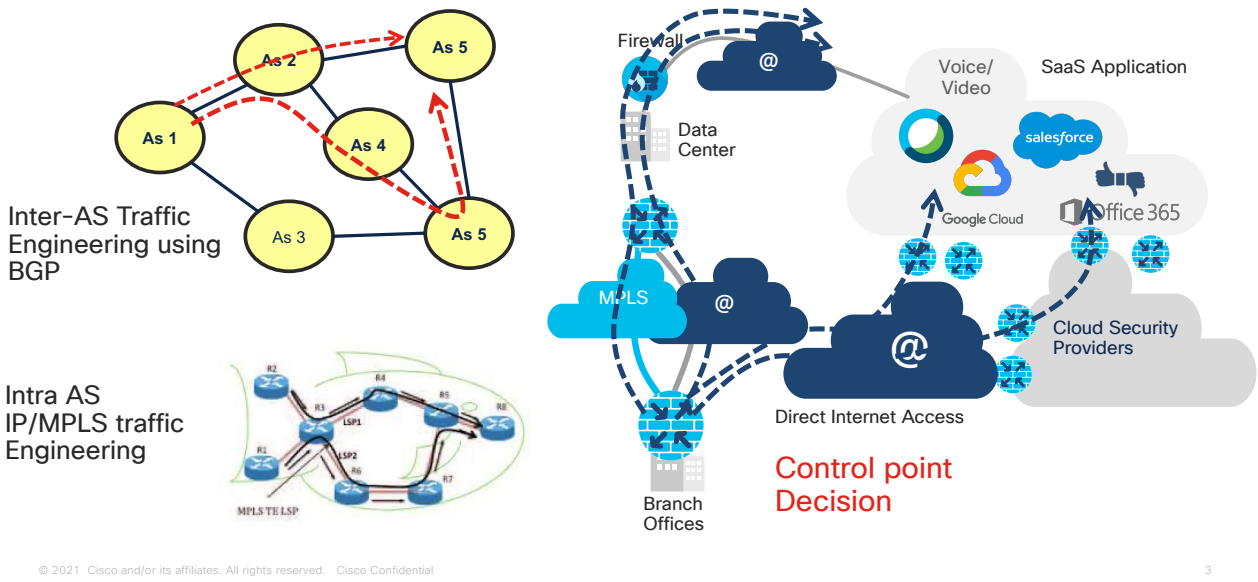

Figure 5: Edge Path Selection in a Predictive Internet

The notion of Predictive Networks for SD-WAN refers to the ability to predict (dark/grey) failures for each of the paths from edge to destination, on a per-application basis.

Important Note: Can the system predict where in the network the issues take place? The objective of such an engine is not to predict the exact location where the issue may take place in the Internet and then use some potential loose-hop routing or segment routing technique to avoid the predicted failed location. **Instead, the engine predicts issues for a given path (per-path modeling) and allows for the selection of an alternate path end-to-end, making the prediction of the root cause not required.**

5 Reactive versus Predictive

As first discussed in (Vasseur J. , Mermoud, Kolar, & Schornig, 2022), reactive and predictive technologies are quite different and highly complementary.

Reactive strategies necessitate first detecting the issue before rerouting traffic onto an alternate path. By definition, this implies that the network has already encountered an issue, and the Quality of Experience (QoE) has been compromised. In instances of total connectivity loss, Keep-alive (KA) mechanisms typically respond within seconds, depending on the configuration. However, if the issue pertains to SLA breaches that demand the use of probes (as seen in Grey Failure [3]), the system must first evaluate the QoE for a certain duration before initiating a reroute, a process that often spans several minutes. While one might employ aggressive timers to expedite convergence time, this can inadvertently instigate oscillations. An overly reactive strategy is likely to induce excessive rerouting, thereby further impairing the user experience. Especially on paths where transient issues are commonplace — a frequent scenario on the Internet — such a method could result in perpetual rerouting and potential path changes, which are highly undesirable even when mitigated using some form of hysteresis.

Furthermore, once traffic has been rerouted, there's no immediate insight into the QoE on the alternate path. In contrast, predictive engines can employ models that allow for testing against new conditions prior to rerouting.

As illustrated in Figure 6 Dark (in red) versus Grey (in grey) Failure, studies across a vast array of networks have highlighted a significant prevalence of grey failures (where user experience is degraded) in comparison to dark failures (complete loss of connectivity). For every grey failure, a reactive strategy would necessitate an SLA assessment over a specified time span, during which an SLA violation is already being experienced, before any rerouting can occur.

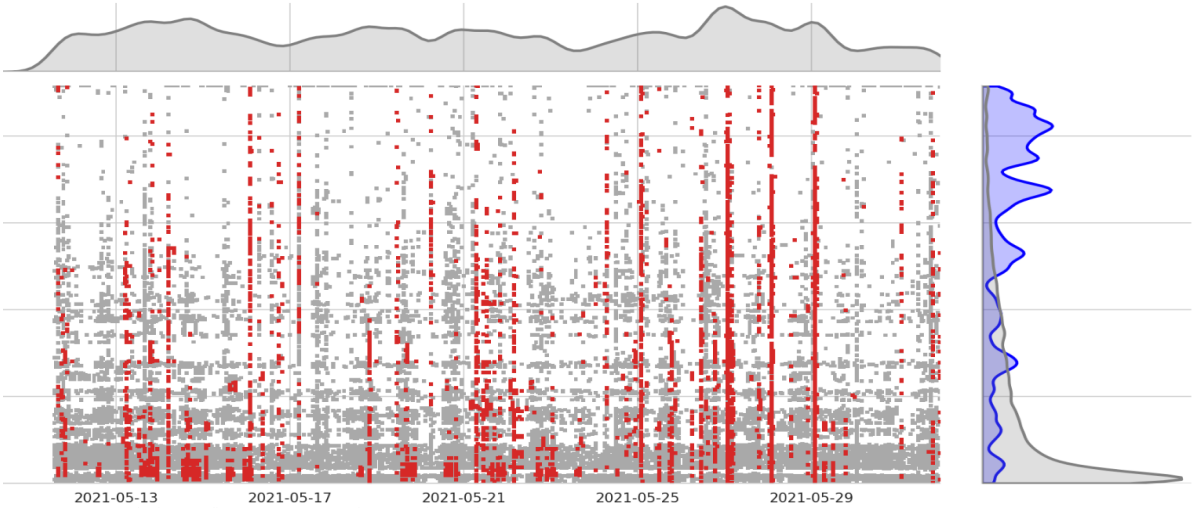

Figure 6 Dark (in red) versus Grey (in grey) Failure

Consider a path across the internet that experiences daily congestion at a specific time. A reactive approach would consistently impact the user at this same time every day. In contrast, a predictive approach could anticipate and circumvent this congestion. While seasonality serves as one potential signal for predictions, other early signs, like fluctuations in jitter, can also be used to forecast grey failures before they materialize.

Figure 7 Illustration of predictive vs reactive protocol for avoiding bad -experience. depicts the difference in user experience between reactive and predictive approaches. The Y-axis represents the SLA violation, with the green line illustrates the ground-truth SLA violation over a path. The blue dashed line indicates the predicted SLA violation based on the predictive method, while the red dotted line portrays how reactive methods estimate SLA violations based on past data. Although a reactive strategy might eventually adapt after the commencement of a grey failure, it will always struggle to fully negate the associated disruptions.

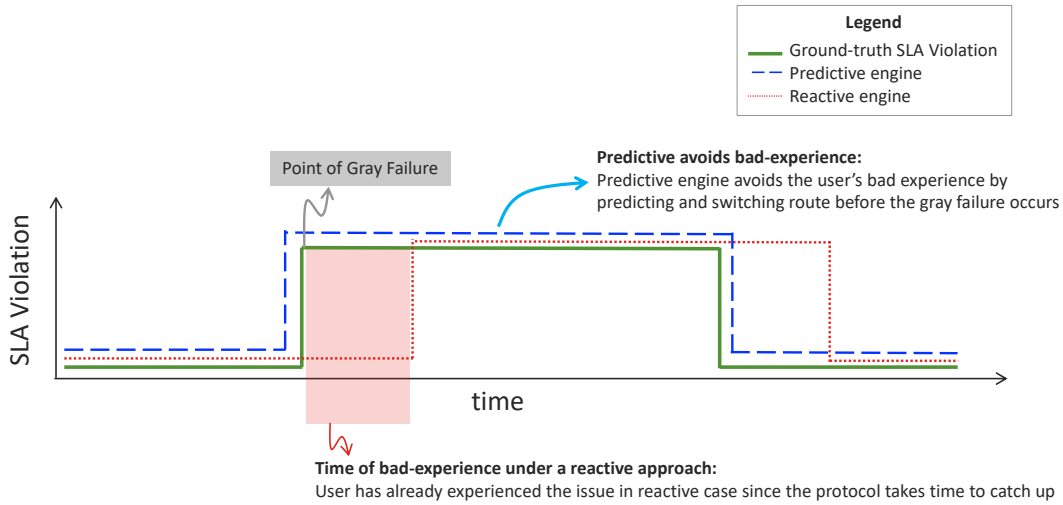

Figure 7 Illustration of predictive vs reactive protocol for avoiding bad -experience.

Moreover, any system predicting events using statistical or ML models must be fine-tuned with specific objectives in mind. Take, for instance, the classic example of predicting events using a classification ML algorithm. ML engineers consistently grapple with the challenge of optimizing their system for either highest precision or recall.

While the goal is to maximize both precision and recall, there exists an inherent tradeoff between them: increasing recall often leads to a decrease in precision and vice versa. Cisco's predictive engine is optimized for high precision, meaning it strives to avoid as many disruptions as possible (with precision ideally nearing 1). Put another way, the objective isn't necessarily to predict every single issue, but to ensure that when an issue is predicted, it almost certainly transpires (i.e., the system seldom produces false positives). This level of precision is vital for facilitating trusted automation in self-healing networks. Moreover, traditional reactive networks—and the Internet at large, for that matter—effectively have a recall of zero, since no issue is ever preemptively identified and circumvented.

5.1 Can we quantify the benefits of Predictive Versus Reactive?

Reactive approaches, by definition, cannot predict the onset of application disruptions. To quantify the benefits of a predictive engine, we introduce the notion of Application Failure (AF). As pointed out, reactive strategies cannot avoid AFs: traffic along the impacted path will experience degraded conditions for as long as the detection is not confirmed (between a few minutes and an hour depending on the conditions) and then only the traffic will be re-directed onto a **(potentially)** non-violating path. Figure 8: Performance of our predictive engine on 11 enterprise networks, for a total of more than 108 000 paths.illustrates (a) the fraction of AFs successfully predicted by current version Cisco's predictive engine and (b) the fraction of how many sessions, session-minutes or users could have been forecasted to have an AF.

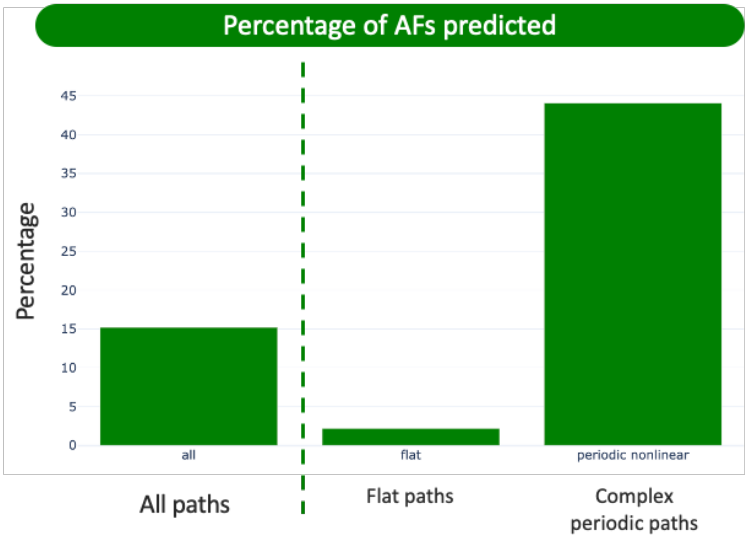

Figure 8: Performance of our predictive engine on 11 enterprise networks, for a total of more than 108 000 paths.

Across all paths, 15% of AFs are predicted correctly. As discussed in details prediction accuracy varies with the nature of the path. For so-called *flat* paths that have only very few issues, the engine predicts only 2% of the AFs, most likely because those are due to very unpredictable mechanisms. However, on a subset of the paths that exhibit a periodic pattern, the model predicts more than 44% of the AFs correctly. This demonstrates the need for predictive methods in situations where patterns of violation exist. Note again that reactive approaches have, by definition, a recall of 0%.

5.2 The Synergy of Predictive And Reactive approaches

Predictive systems will never be able to foresee and forecast all issues; therefore, a reactive approach is necessary. If an issue is predicted, traffic will be proactively rerouted onto an alternate path. However, if the issue isn't foreseen, the reactive mechanism will be activated.

This demonstrates how both mechanisms operate harmoniously and are entirely complementary. Since the system is optimized for high precision (ensuring no false positives), all actions proactively initiated by the predictive mechanism will have a beneficial impact. Meanwhile, any unforeseen issues will be addressed in the same manner as in current networks.

6 Vision or Reality? Results

Cisco's predictive engine has been in production in 100s of networks around the world, doing real-time predictions for months and has been proven to perform accurate predictions highly improving the overall network SLO and application experience. Although the details of the exact architecture, telemetry (and technique for noise reduction), algorithms, training strategies are out of the scope of this document, it is worth providing several examples of the overall benefits that a predictive system can bring.

Although the system continues to evolve and improve using new sources of telemetry, algorithms and tuning of many sorts, it is worth providing some high-level overview of the performance one can expect (with concrete examples) from such predictive technologies.

6.1 Short term predictions

Let us start with a few examples. Figure 9: number of minutes of application SLA violation (red) and number of minutes of SLA violation a predictive system would have avoided (real values in an existing network) shows the overall number of minutes with SLA violation observed in a 30-day period on a network (in Red). Next to it are the number of minutes of SLA violation that would have been avoided ("saved") using the predictive engine. In this case, the predictive engine

managed to accurately predict such grey failures (application SLA violation) but also find an alternate paths free of SLA violation in the same network (without adding any additional capacity).

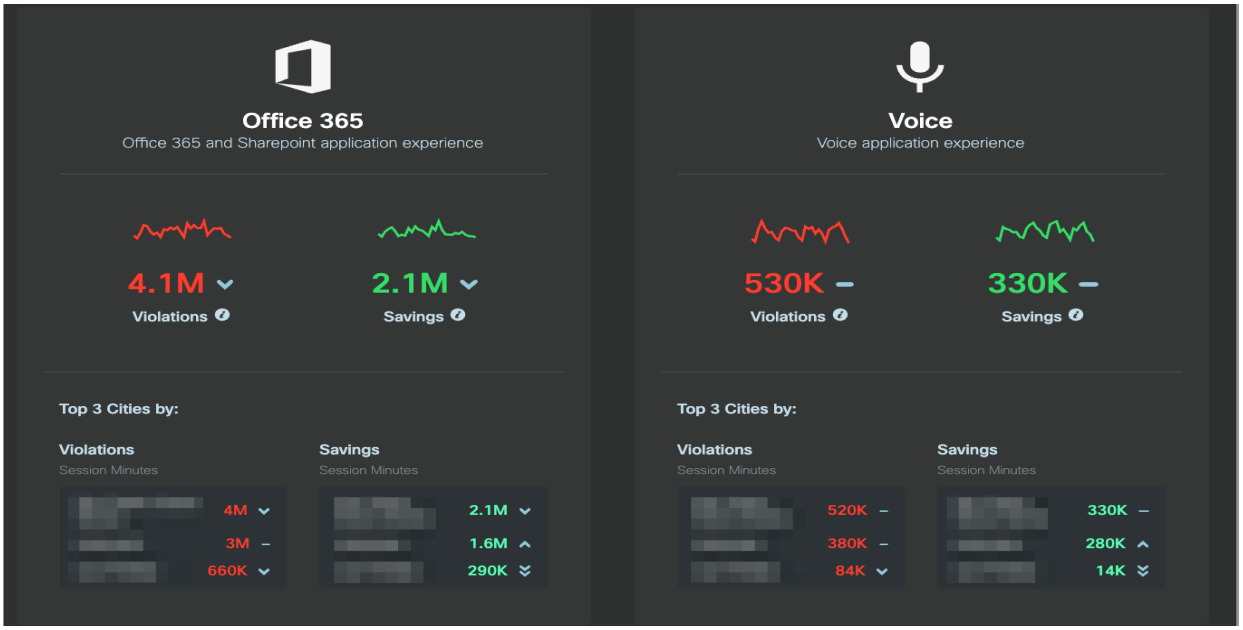

Figure 9: number of minutes of application SLA violation (red) and number of minutes of SLA violation a predictive system would have avoided (real values in an existing network)

Note that the ability to find an alternate is highly governed by the network topology and its dimensioning. In many circumstances, it was shown that the engine correctly predicted issues but there was no alternate path available in the network that could be used.

For the sake of illustration, let us explore examples of such predictions (i.e., SLA violation) that were correctly predicted. The next set of figures show when a failure was predicted (see green dots on the timeline). Various time series show after the predictions the loss, delay and jitter. In ocean blue is the default path programmed on the network, in the dark blue color is the path recommended by the predictive engine, thus validating that the prediction was indeed correct.

In the first example Figure 10: Prediction of SLA violation because of packet loss on a path between Costa Rica and Malaysia: 90% of SLA violation avoided many minutes of traffic (11,000 minutes of voice traffic) with SLA violation could have been avoided (green) for traffic sent along Business Internet path (ocean blue) by proactively rerouting traffic onto an existing bronze internet path (a priori with less strict SLA) (dark blue).

Figure 10: Prediction of SLA violation because of packet loss on a path between Costa Rica and Malaysia: 90% of SLA violation avoided

Figure 11: Prediction of high packet loss spikes on an MPLS path shows a prediction of packet loss spike (way before they take place) along a short-distance MPLS paths resulting in 82% of SLA violation over a 30-day period.

Figure 11: Prediction of high packet loss spikes on an MPLS path

Figure 12: Prediction of sporadic packet loss is another example of prediction of a sporadic packet loss (17%) in Australia.

Figure 12: Prediction of sporadic packet loss

The Predictive Engine recommendations can be inspected in the Thousand Eyes UI in the WAN Insights (WANI) section. Below we discuss two examples of such recommendations using the WANI dashboard.

The first example is a routing recommendation for **Salesforce** application using direct internet access (DIA) circuits. At top level, a summarized tile is used to convey to the network admin the most important information such as the application, SD-WAN site Id, location information, current and recommended circuits and forecasted quality improvement:

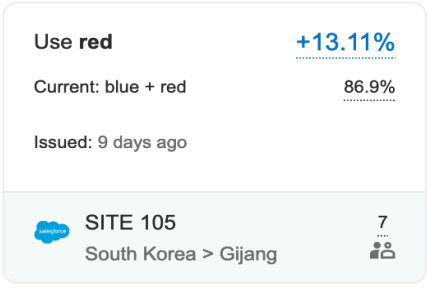

Figure 13 - Salesforce Recommendation ~ Top Level View

In this example the routing recommendation states that **Red DIA** circuit should be used instead of load balancing on **Blue and Red** in order to gain a 13% QoE improvement for the Salesforce application.

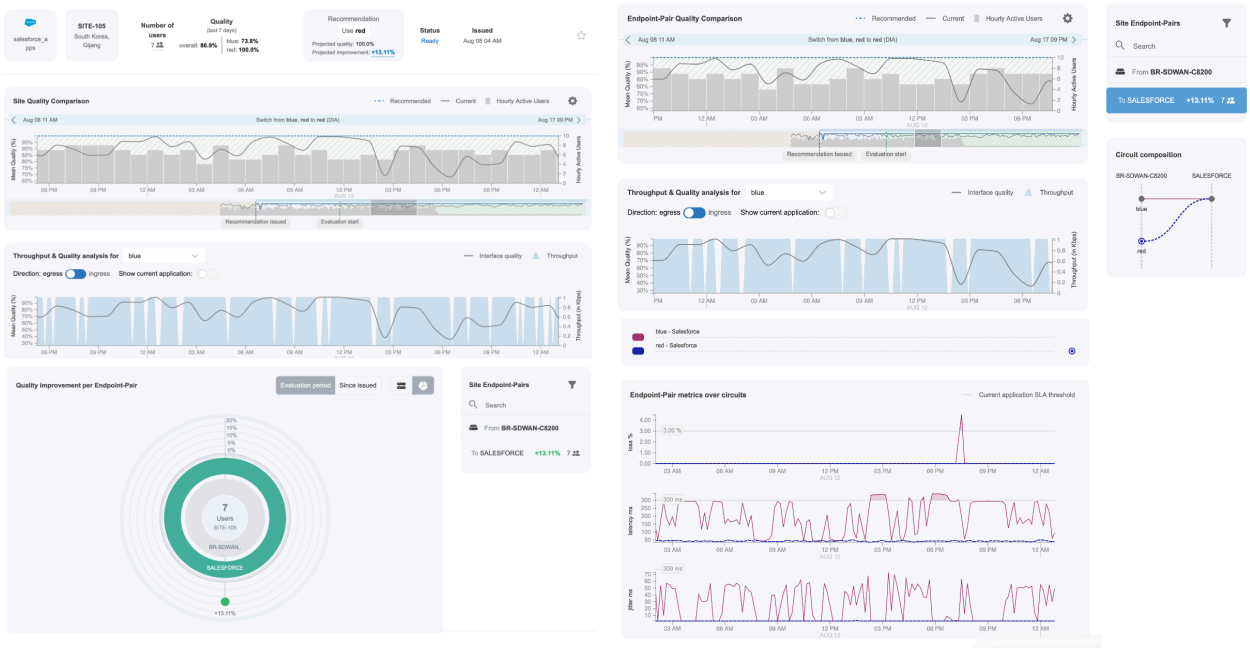

Figure 14 - Salesforce Recommendation ~ Detailed View

Figure 14 - Salesforce Recommendation ~ Detailed Viewshows a more detailed view of the recommendation. The *Site Quality* and the *Endpoint-Pair Quality Comparation* graphs allow the user to inspect the historic quality metrics for the current (grey line) and recommended (dotted blue) circuits over the last 30 days.

Here we notice the QoE metric for the current circuit selection (grey line) experiences frequent degradations, while the QoE score over the recommended circuit (blue dotted line) remains stable. At the endpoint pair level view (right side of the diagram), we can further inspect the loss, latency and jitter measurements collected from the actual network devices and determine the cause of the QoE degradations, which in this case can be attributed to high latency and jitter (bottom right). The same pattern can be observed over multiple days.

A second example is for a recommendation related to a group of Office 365 applications running over SD-WAN tunnels.

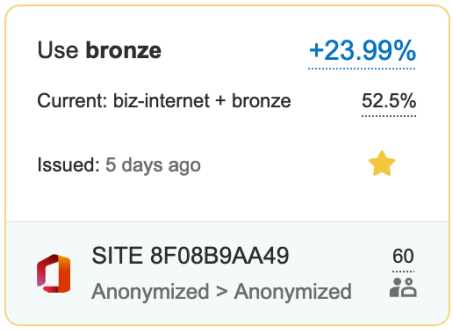

Figure 15 - Office 365 Recommendation ~ Top Level View

In this second example the routing recommendation states we should use the **Bronze** circuit and avoid using **biz-internet,** in order to gain 23.99% QoE improvement for M365 apps.

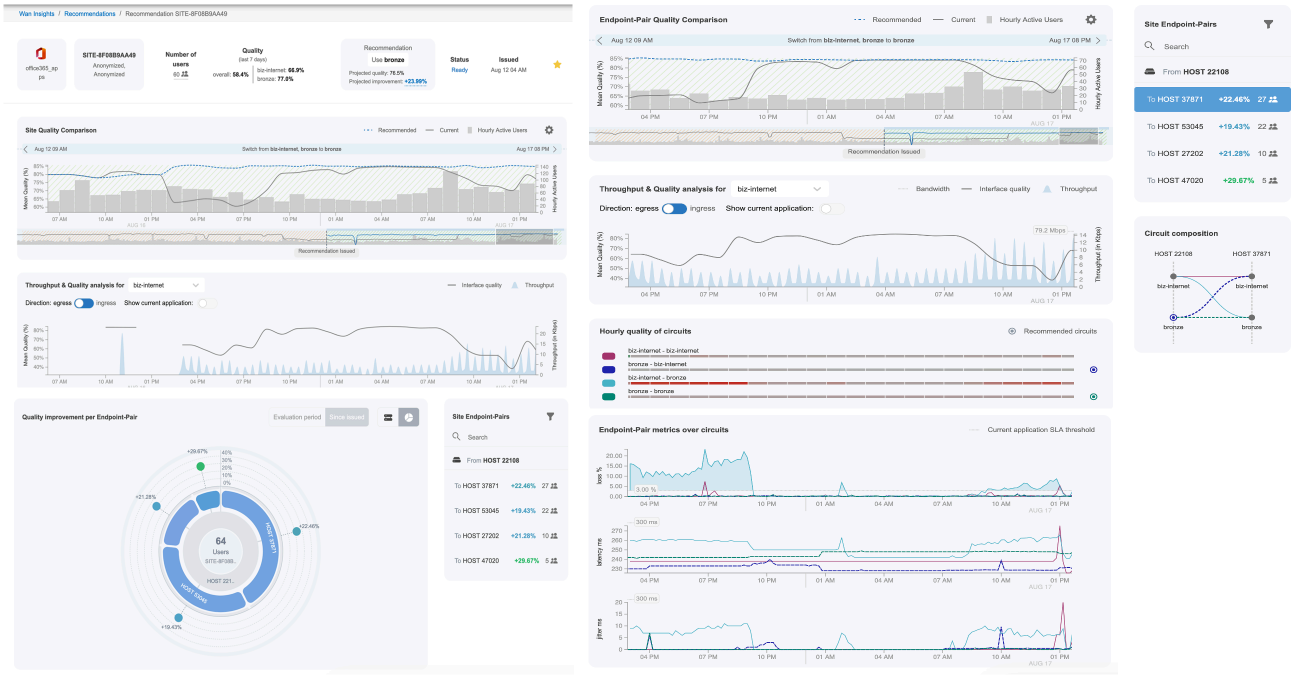

Figure 16 - Office365 Recommendation ~ Detailed View

Figure 16 - Office365 Recommendation ~ Detailed Viewshows a more detailed view of the recommendation. Similarly to before, we notice the QoE metric for the current circuit selection (grey line) experiences dips, while the QoE score over the recommended circuit (blue dotted line) remains mostly stable.

Inspecting the endpoint pair level view (right side of the diagram), shows that some of the SD-WAN tunnels over the biz-internet circuit are frequently experiencing packet loss upwards of 10% likely causing QoE user impact.

The number of such successful predictions is very high and a very few examples have been provided in this document for the sake of illustration.
Even though all failures cannot be predicted, any single prediction means that an issue is being proactively avoided, while all other (unpredicted) failures are being handled by the reactive system, making the combination of both approaches a huge step forward for the Internet.

Are accuracy and efficiency of predictions similar across networks and time? No, as expected. First, each network has its peculiarities in terms of topology (and built-in redundancy), traffic profiles, provisioning, and access types to mention a few. A predictive engine may then exhibit different levels of efficiency, as expected. Furthermore, the objective is not just to Predict but also to find some alternate path free of SLA violation. Consequently, the built-in network redundancy (availability of alternate paths) will be a key factor. A very interesting fact that has been observed is that predictions do vary significantly over times as the Internet/Networks evolve (e.g., failures, capacity upgrades, new peering) requiring constant learning and adjustment. Figure 17: Number of minutes of traffic with SLA violation avoided thanks to accurate predictions on six worldwide networks shows the number of accurate predictions and ability to find alternate path (with measures such as the number of minutes of traffic saved from SLA failures) and their variation other times, for multiple regions of the world and across multiple networks.

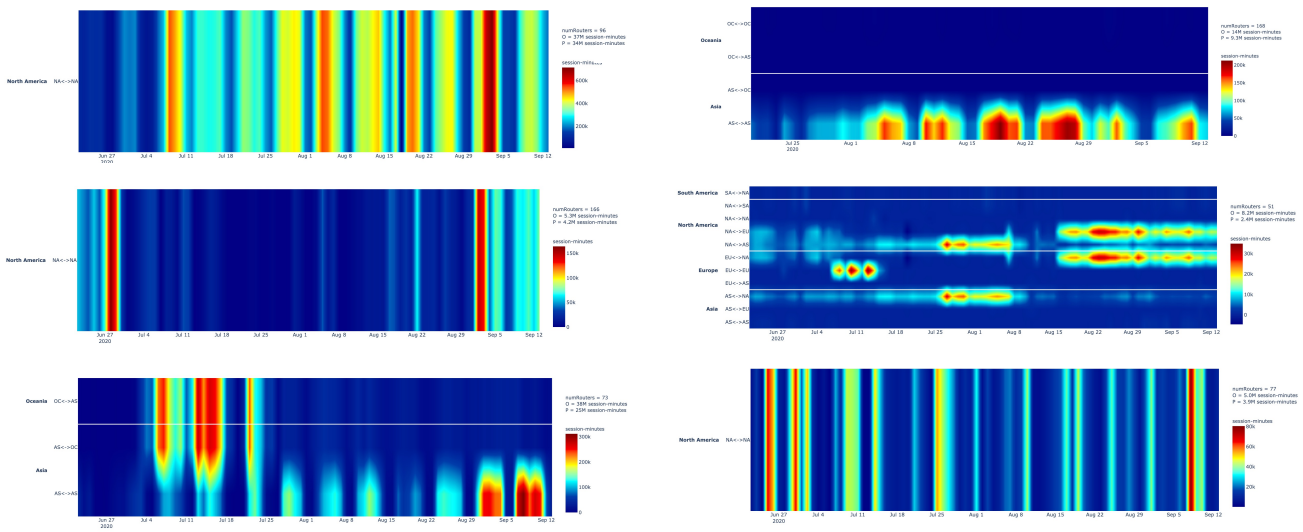

Figure 17: Number of minutes of traffic with SLA violation avoided thanks to accurate predictions on six worldwide networks

The Y-axis shows the regions of the world and the X-axis the number of saved minutes of traffic with SLA violations varies over time. It can be noticed that the amount of traffic saved varies significantly across networks and over time. On the top left corner, a network where the number of failures accurately predicted and avoided varies from low the high across all regions. On the top right, another network where most of the failures avoided are located in a given region and tends to exhibit some form of periodicity. We can observe yet another pattern on the bottom left network where at some point, most of the avoided failures were in a given region and later in another region. It is also worth noting that predictions are constantly adjusted thanks to continuous learning. Indeed, the Internet and other SP networks are highly dynamic and experience failures due to several reasons such as new peering agreements, movement of SaaS applications across the Internet and evolution of traffic loads.

As often, there is no one-size-fits-all algorithm capable of predicting (grey) failures. Each algorithm has specific properties related to the type of telemetry used to train the model, the forecasting horizon (itself coupled with the availability of telemetry) and of course the overall efficacy.

For the sake of illustration, Figure 18: Variability of prediction accuracy per region per path for a given algorithm. shows a performance accuracy metric for multiple paths in the world with different characteristics. The metric for measuring the accuracy is outside of the scope of this document but the point is to show that accuracy varies (in this particular example, a low value is indicative of higher performance).

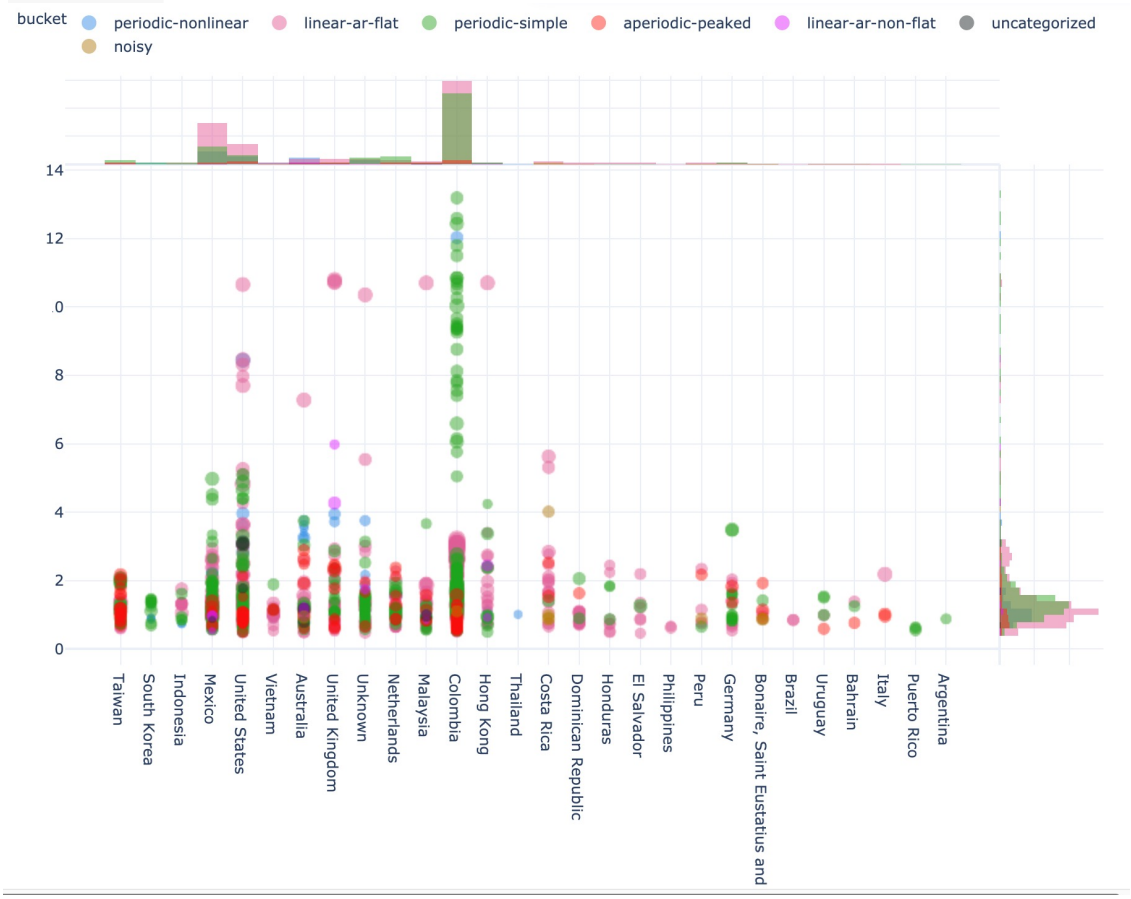

Figure 18: Variability of prediction accuracy per region per path for a given algorithm.

Figure 19: Relationship between forecasting accuracy and path characteristics for a given algorithm provides another view of such variability of the predictive accuracy; one can observe the relationship between forecasting accuracy and some properties of the path such as the entropy, level of periodicity and many other path characteristics.

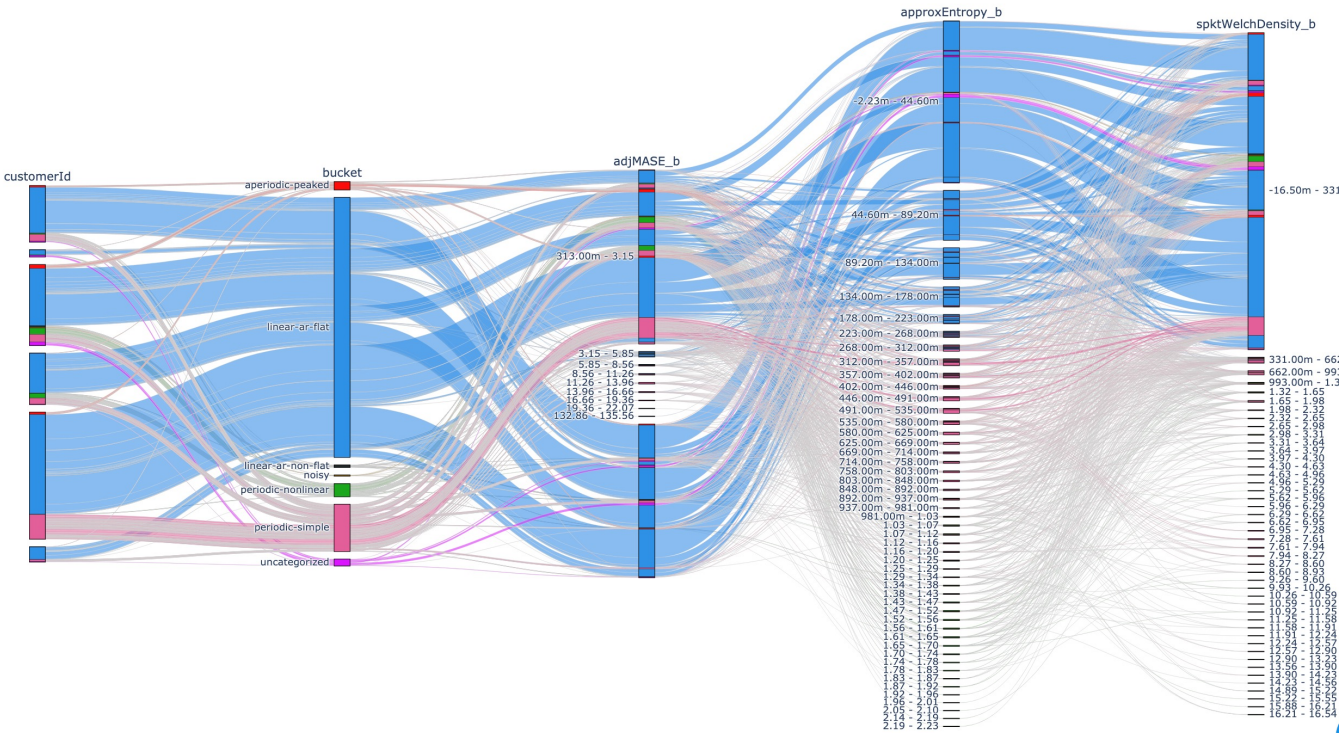

Figure 19: Relationship between forecasting accuracy and path characteristics for a given algorithm

6.2 A deeper-dive into some results

Let us now have a deeper look at some of the prediction performance metrics for a *given algorithm*. A disclaimer: such numbers must not be considered as the definitive truth since they constantly evolve and they are algorithmic-specific, but provided here for the sake of illustration.

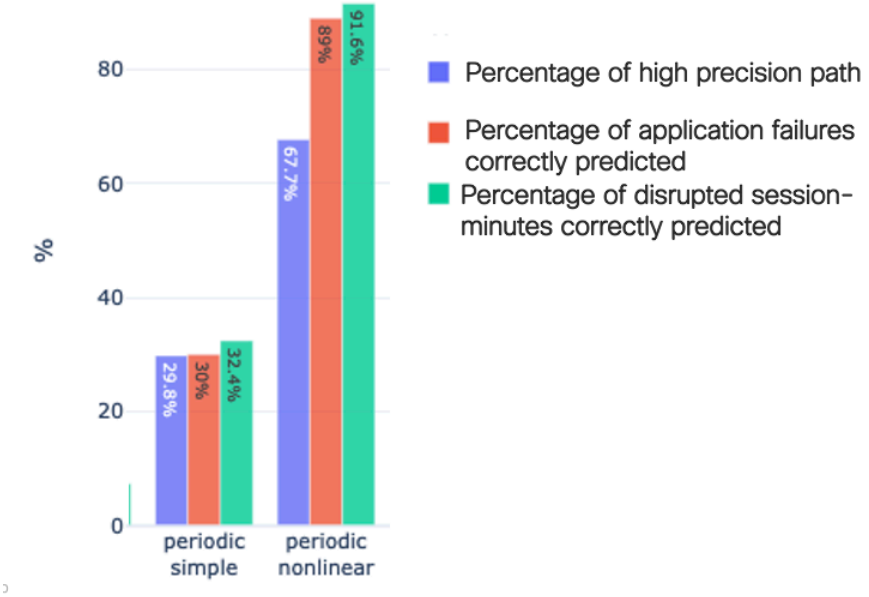

Figure 20: Percentage for High Precision path, application failures and disrupted session minutes correctly predicted for different categories of paths

Figure 20: Percentage for High Precision path, application failures and disrupted session minutes correctly predicted for different categories of paths shows remarkable performances obtained for two path categories (called "Periodic simple" and "Periodic non-linear"). For a detailed discussion on path categories, refer to the study on time-series categorization *(Dasgupta, Kolar, & Vasseur, Jan 2022)*. For the periodic nonlinear category, 67% of the paths had at least 20% of Recall and 90% of Precision (called high precision paths), 91% of disrupted session minutes were correctly predicted (very high recall) and 89% of the application failures were predicted (recall for AF metrics). For this particular algorithm, lower performance was obtained for the recall, still allowing for the avoidance of a number of issues in the network thanks to accurate predictions.

Figure 21: Illustration of a predictive algorithm Illustrates the ability of the algorithm to anticipate and predict issues.

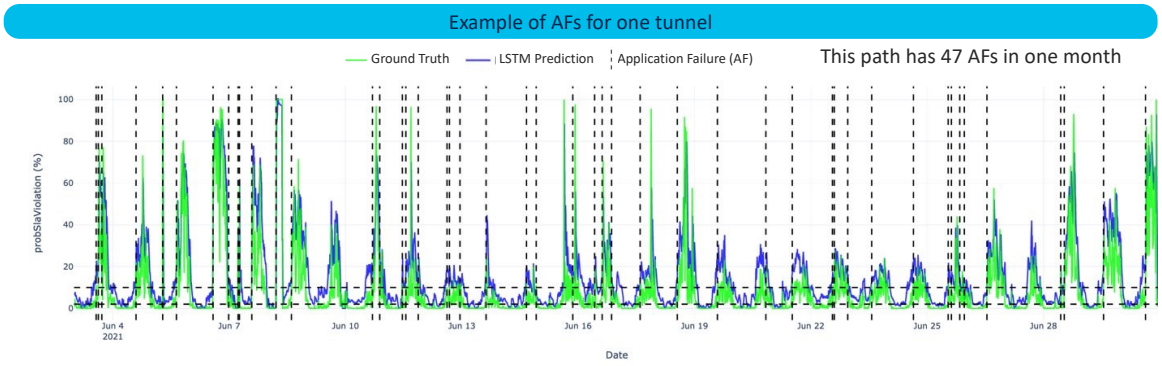

Figure 21: Illustration of a predictive algorithm

Precision and Recall may even be path-specific for a given category. Figure 22: Precision-Recall per-path using an LSTM algorithm for a specific category of paths is an illustration of the Precision/Recall distribution for a number of paths for a given (LSTM) algorithm.

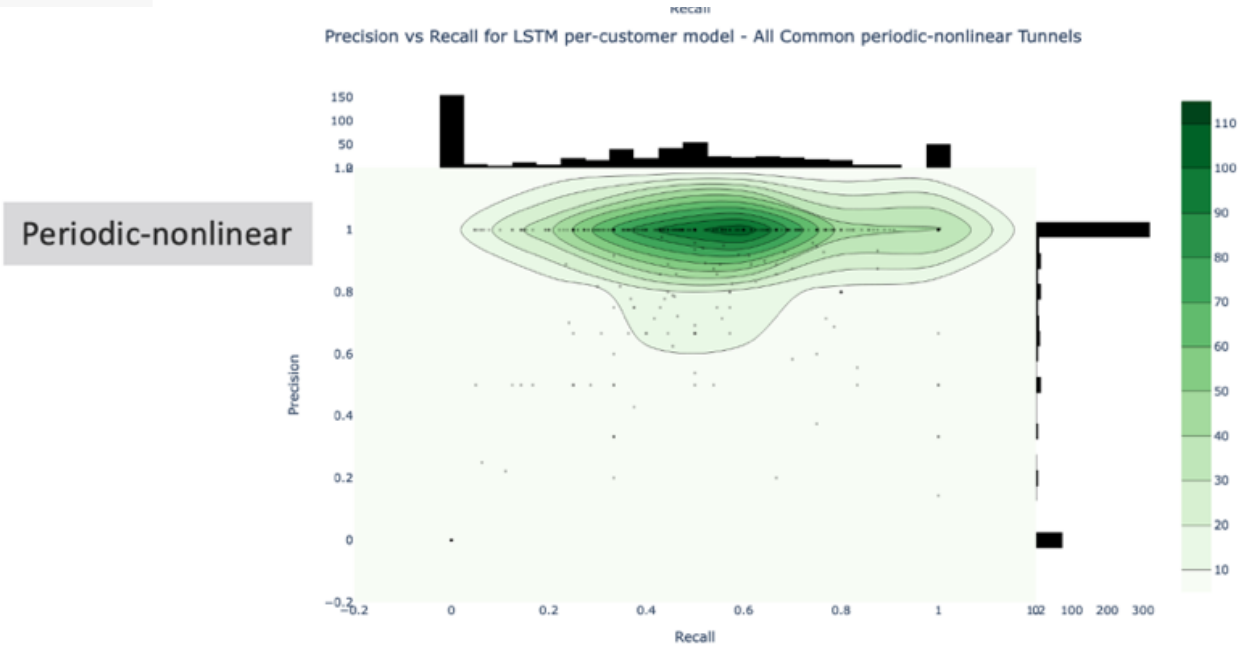

Figure 22: Precision-Recall per-path using an LSTM algorithm for a specific category of paths

6.2.1 Long term predictions

In contrast, long-term predictions are made by other types of models to predict the probability of failures in upcoming days, weeks and even months. Because of a different forecasting timeline, such predictions may then be used by a recommendation system to suggest configuration changes in the networks. Note that this is in contrast with short-term predictions that work in-hand with full automation. Long-term prediction systems may take additional constraints into account such as the duration of validity of such predictions. Indeed, in absence of automation, it is inconceivable to make too many recommendations that would require manual interventions (configuration changes) by the network administrator.

The next figures show results obtained using a long-term prediction algorithm on dozens of networks for multiple applications. As always one must define the metrics of interest. In this case, in order to evaluate the impact of failures we use a metric called the MPDU (Mean across all time of the maximum number of users in a day; number of users are sampled every 1h).

Figure 23: Long-term predictive algorithm reducing the max number of impacted users by 10,000 (from 40K to 30K) summarizes the overall benefits for 61 networks showing that the maximum number of impacted users (measured per hour) at peak was reduced by 25% thanks to long term predictions with a very high accuracy.

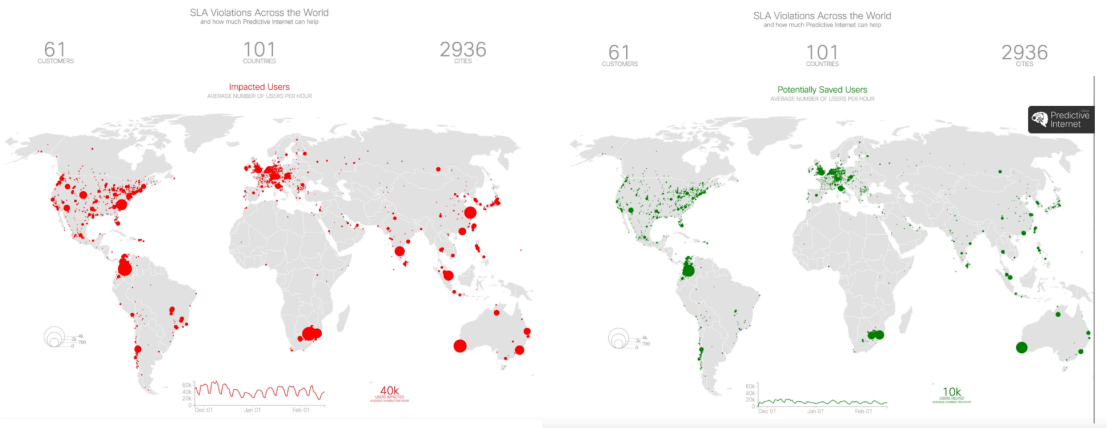

Figure 23: Long-term predictive algorithm reducing the max number of impacted users by 10,000 (from 40K to 30K)

Let's now have a deeper dive into the results:

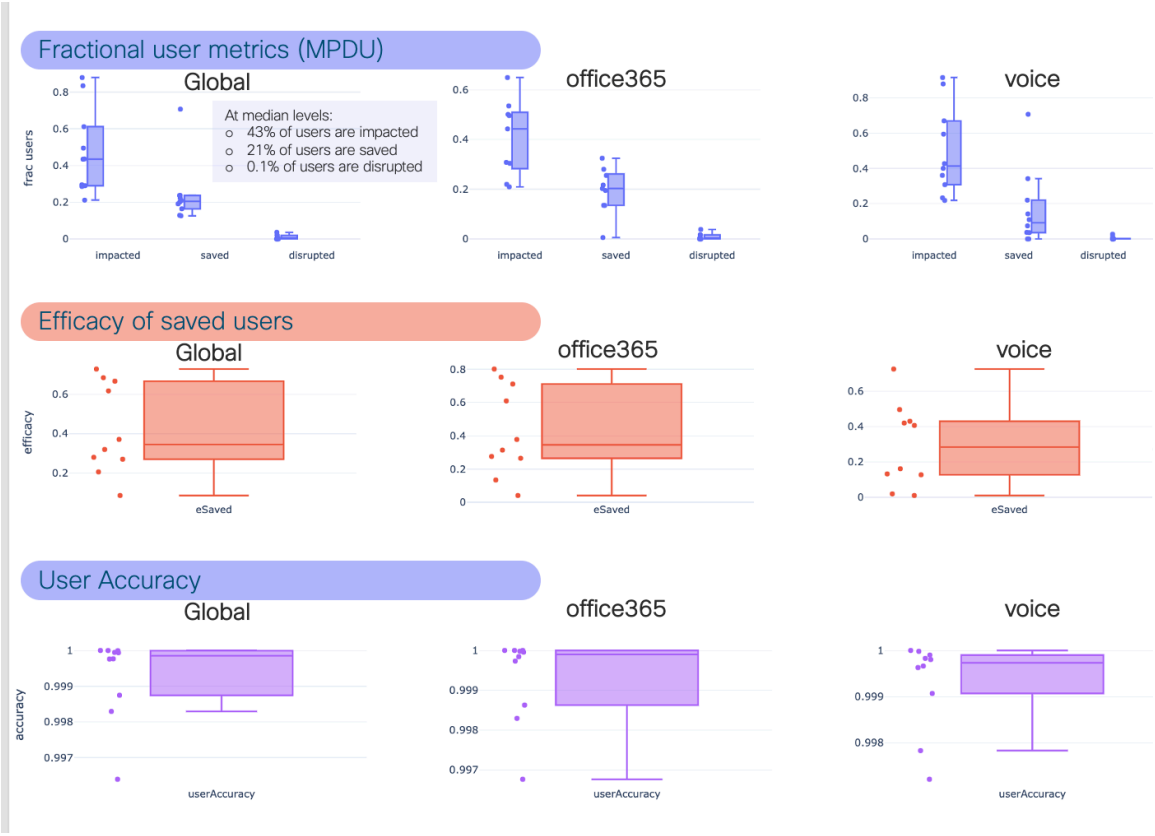

Figure 24: Various performance metrics of a long-term prediction algorithm

Figure 24: Various performance metrics of a long-term prediction algorithm shows a number of interesting performance metrics for all applications (Global), Office365 and Voice. The results speak for themselves. The first plot shows the percentage of users at peak that were impacted by failures (i.e., SLA violations), saved (thanks to the correct predictions of the algorithm, users were not impacted), and disrupted (the user experiences were worst on the recommended path than the default path – the percentage of time within SLA was less than 90% of the time). Box plots are used to illustrate the distributions across a number of sites and networks. The second plot illustrates the efficacy (percentage of impacted users that were no longer impacted thanks to the correct prediction). **One can note also the remarkable accuracy of the algorithm. Moreover, the average lifetime of those predictions exceeded one month in the majority of the cases.**

As illustrated in Figure 24: Various performance metrics of a long-term prediction algorithm, as for short-term predictions, efficacy varies over time, which once more, shows the need for constant learning so as to adapt to the high dynamicity of the networks. The evolution of efficacy is shown for 10 networks. It can be observed that efficacy is constantly high for some networks whereas it does vary for over networks. Still, it can be shown that the overall performance is significant.

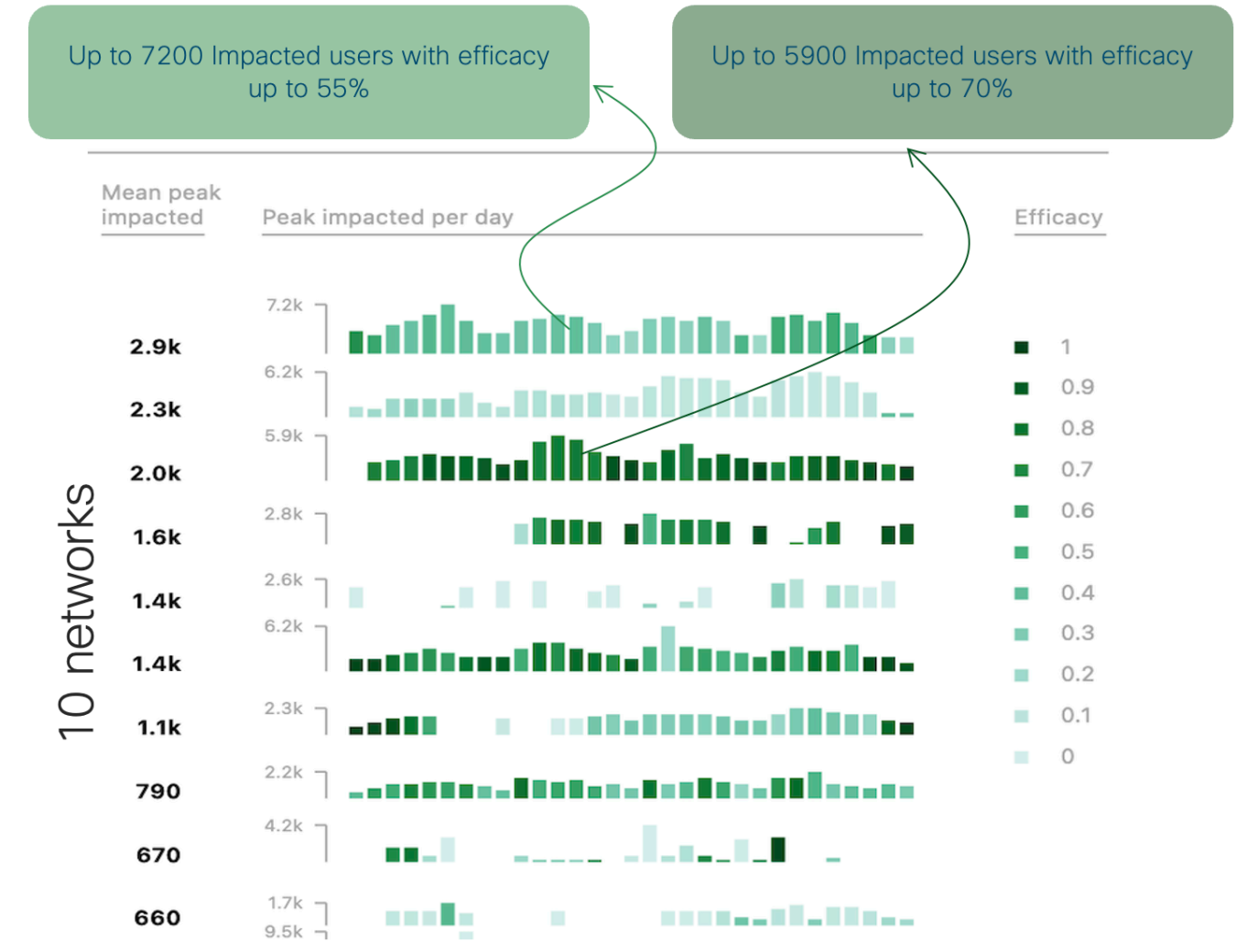

Figure 25: Variability of prediction efficacy for a set of paths

Let us now analyze the performances of the prediction algorithm using additional metrics: quality (percentage of time during which SLA is met) for the default path (paths used by the network according to the SD-WAN policy) and the

recommended path (recommended paths by the predictive engine), measuring also the total number of users impacted (with some SLA violation), for all applications.

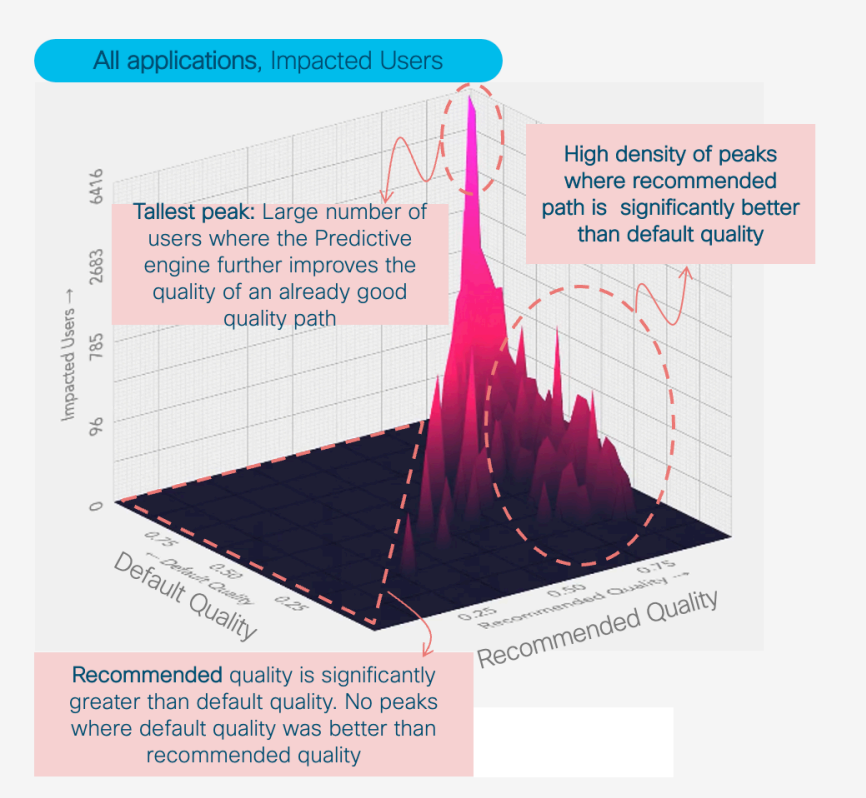

Figure 26: Evaluation of the degree of user experience improvements of recommended paths compared to default paths

Figure 26: Evaluation of the degree of user experience improvements of recommended paths compared to default paths provides several interesting insights. For example, there are peaks (high number of impacted users) with high efficacy where the default quality was relatively good. But there are also may peaks showing situations where default quality was low and efficacy is high; thus denoting that such predictions help many users when default quality is low.

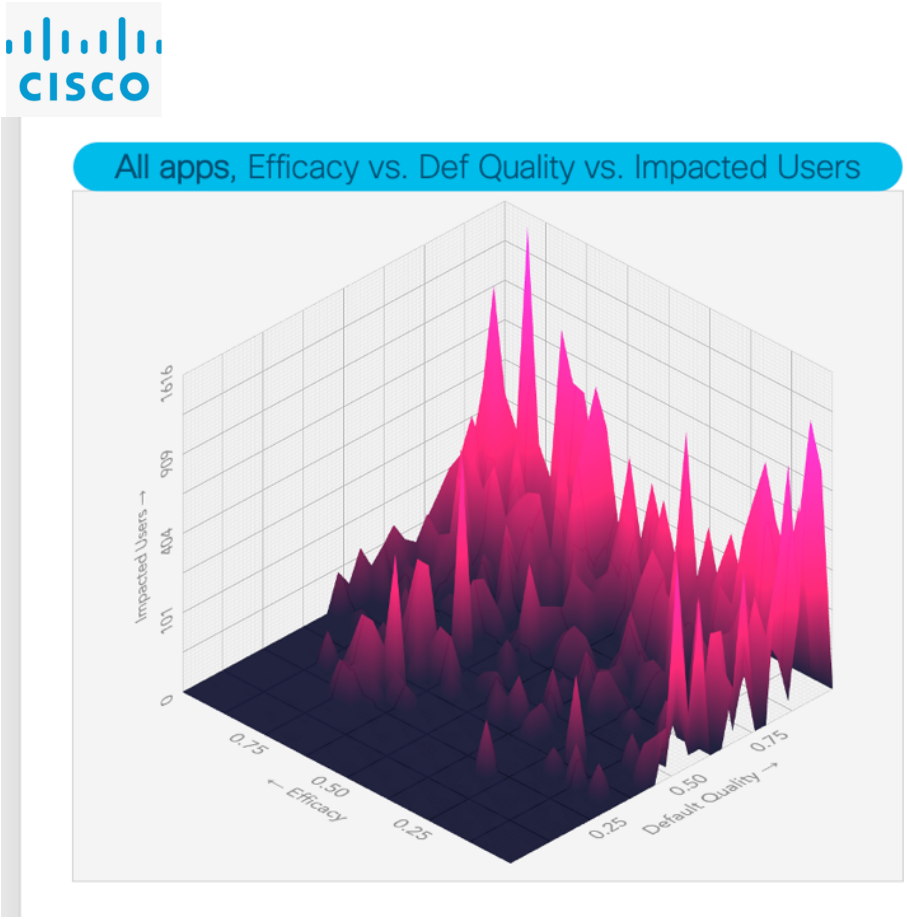

Figure 27: 3D plots showing Default quality, Efficacy and impacted users

7 Towards a Predictive Internet with Predictive BGP?

In regard to Internet routing, BGP has seen impressive deployment. Its path selection metrics are relatively simple, primarily based on the number of traversed ASes. These metrics do not consider network performance KPIs (such as loss, latency, jitter, and throughput) which are crucial for popular applications like collaboration tools, gaming, VR, and streaming services.

Maintaining a high Quality of Experience (QoE) becomes challenging when traffic consistently passes through multiple ASes before reaching its destination, especially when depending on a protocol that's unaware of the underlying network performance. To address this, Network Administrators often create intricate BGP policies that are applied to peering connections. These policies, which manipulate common BGP route attributes like local preference, MED, and path length, are carefully crafted to influence the BGP route selection algorithm and avoid paths that cross underperforming transit circuits or ASNs. Additional factors, such as the nature of the peering relationship, transit costs, and capacity constraints, add further complexity.

This process is largely reactive, with issues typically identified based on user complaints. Resolving these issues can be time-consuming, leading to extended periods of service degradation. The success of these resolutions often depends on the Network Administrator's expertise in applying the appropriate mitigation techniques. Manual adjustments to BGP policies can introduce errors. In fact, there have been several instances in recent history where incorrect BGP policy updates caused widespread Internet outages.

As noted earlier, Predictive Networks possess the ability to learn, predict and plan. Earlier sections discussed the application of this technology in SD-WAN. Predictive Networks technology could enhance inter-AS traffic management across the Internet. Employing a closed-loop automation system would enable the implementation of recommendations and adjustments to BGP policy configurations ahead of anticipated events, allowing Service Providers to proactively and automatically manage their peering infrastructure.

Figure 28 Predictive Inter-AS Routing with BGP illustrates how Predictive Networks could be applied to Inter-AS traffic: automatic peering point discovery, dynamic setup of path performance monitoring for high-traffic destinations, and utilization of a Predictive Networks system (PNS) to forecast network performance issues in advance. This system could then provide BGP routing recommendations and adjustments. A closed-loop automation system would dynamically modify the BGP peering policies based on the prediction recommendations.

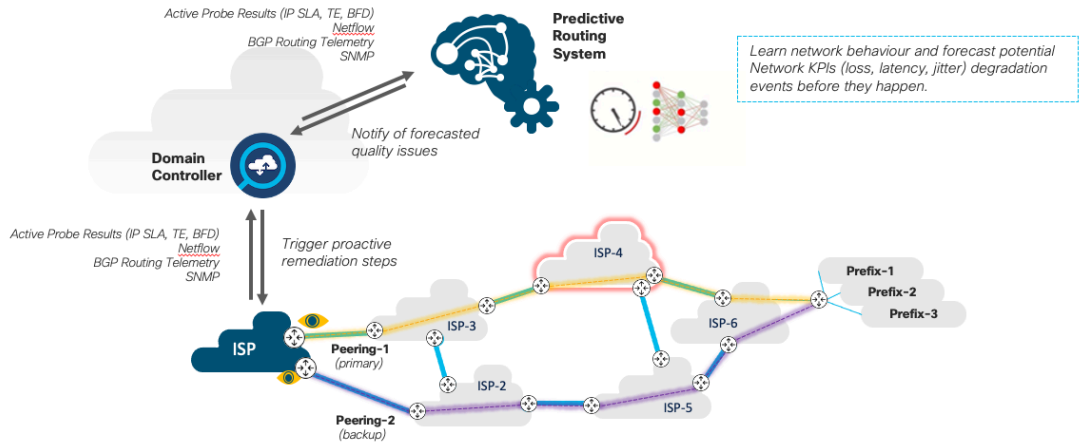

Figure 28 Predictive Inter-AS Routing with BGP

The details of such an implementation are outside of the scope of the document, but the following figures illustrates such a per-prefix prediction:

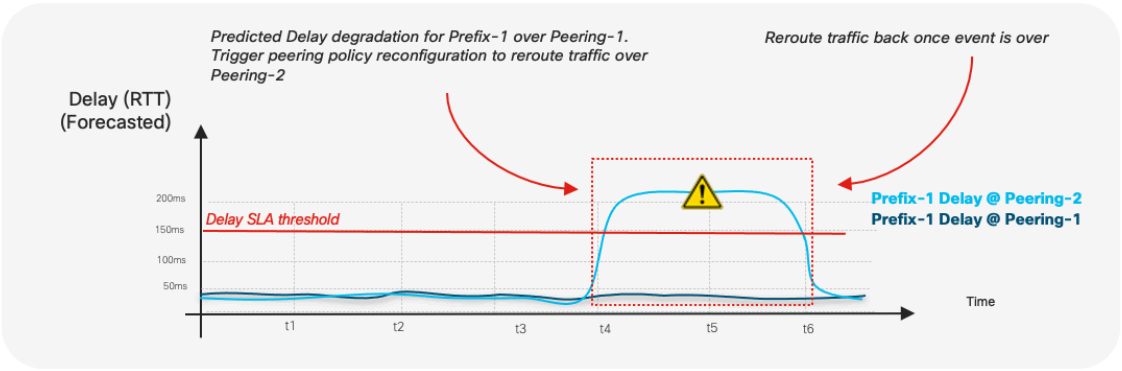

Figure 29

Another component would then take the routing recommendation as input and determine the appropriate configuration changes required for BGP peering policies. Various techniques could be used to engineer egress or ingress traffic paths over specific peering points. Actions such as manipulating the values of BGP attributes, such as Local Preference or route Weight, are used when egress traffic engineering is needed. Meanwhile, techniques based on manipulating MED or the length of the AS Path (path prepend) may be used to influence the path of ingress traffic.

8 Future Applications of Predictive Networks

Predictive Networks technologies has the promise to apply to a broad range of Networking areas: as illustrated in Figure 30 Applying Predictive Networks Technologies to other Networking Areas (SASE, Hybrid work), Predictive Networks could be used to predict the SASE PoP and best path to the PoP in order to guarantee traffic SLA. The ML model would then predict which of the paths is most appropriate instead of selecting the closest PoP according to some (geographic-based) distance or based on periodic probes.

Consider the hybrid work use cases where a user connects to a corporate network from home, potentially using multiple paths (like a local Service Provider via home WiFi or 4G from a laptop or home router). The user might also use a VPN to access a corporate site. In this scenario, traffic can be routed over different physical interfaces, and it can also be sent through a VPN tunnel or not. Using Predictive Networks technology would allow for learning from past experiences, building a model capable of predicting which traffic should be directed over which path. This predictive approach could help avoid the common issues associated with remote work disruptions when working from home.

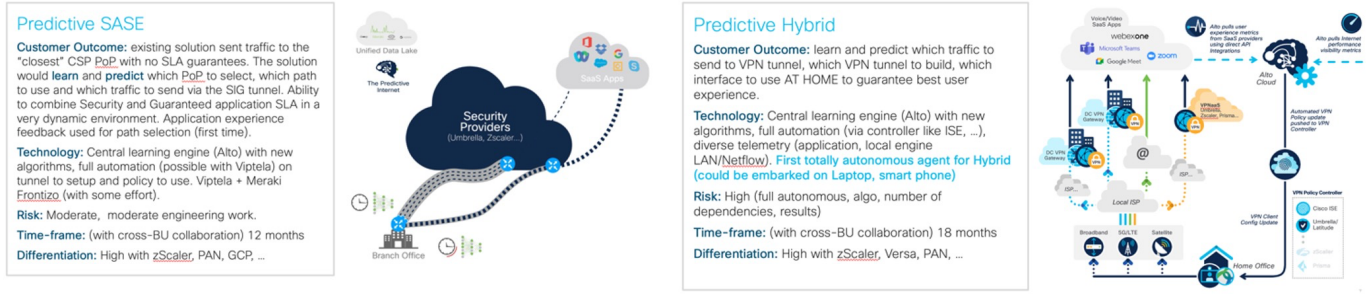

Figure 30 Applying Predictive Networks Technologies to other Networking Areas (SASE, Hybrid work)

9 Real-Time Predictive Networks

Predictive Networks technologies could be utilized to implement both short-term and long-term predictions. This leads to proactive action, helping to prevent failures likely to occur in the near future (e.g., within a few hours to weeks).

Could such a system support real-time predictions and initiate proactive actions within an extremely short time horizon of a few seconds or hundreds of milliseconds? This is certainly a promising and interesting path forward that would necessitate various adaptations to the current approach:

- Use of local inference: While prediction models might still leverage the massive amount of telemetry data in the cloud, the actual inference would need to be local (on-premise). This demands models that are lightweight – a very reasonable constraint. The telemetry used to fuel the model at inference time would be processed in almost real-time, which contrasts with approaches where the forecast operates under different time horizons of hours or weeks.
- Telemetry differences: The telemetry data could be less aggregated, which is beneficial in preserving the original signal. This signal, which likely has a higher predictive power than aggregated signals collected in the cloud, is discussed in detail in (Vasseur J. , Mermoud, Magendie, & Raghuprasad, 2023).
- Complementing reactive approaches: This predictive method could be complemented by reactive strategies if a prediction turns out to be incorrect. For example, if traffic is rerouted onto an alternative path in anticipation of an issue on the primary path, the networking gear could detect the incorrect prediction and immediately revert to the original route. Such a mechanism is less suited for centralized operations.

10 The arm for Self-Healing Network: coupling Prediction with Automation

Without a doubt Automation has always been the holy grail, allowing networks to make remediation actions automatically, at scale, thus improving networks' SLO. Still several roadblocks have been in the way, for a number of reasons. The ability to accurately predict may very well allow for a progressive use of trusted automation, especially for issues the system is capable of predicting. A number of mechanisms have been studied so as to allow networks to make use of predictions with user feedback thus making self-healing networks a reality.

Careful consideration has been given to closing the loop based on predictions.

The first step was to build "trust." Prediction accuracy became a central aspect of this process. One must consider the implications of a prediction system that makes network configuration changes, impacting traffic when predictions are incorrect. This relates to designing the algorithm with the known trade-off between Recall (the percentage of issues being predicted) and Precision (accuracy of the prediction). It's essential to remember that the current Internet operates entirely reactively, so the Recall is effectively 0. To build trust, the design choice was made to optimize Recall while aiming for near-perfect precision (close to 1).

Thus, the first commercial implementation suggested SD-WAN policy changes without taking any proactive actions. This approach allowed users to assess the system's efficacy. In the second phase, users could review all recommendations and decide whether to implement them on the SD-WAN controller automatically, without manual intervention.

A potential third step may be to allow users to configure recommendations for which the system could autonomously initiate actions, while diligently monitoring its prediction performance and the resulting impact on traffic QoE. If a decline

in efficacy is observed, the system could disengage from making predictions for that specific category and notify the network administrator.

11 Conclusion

After 30 years of developing advanced technologies aimed at improving overall network availability, Predictive Networks Technology has showcased the network's ability to Learn (using statistical, ML, and AI models), Predict and Plan. Thanks to numerous large-scale experiments, a variety of statistical and ML-driven models have demonstrated the capability to forecast future events and take proactive measures. Both short-term and long-term predictions for several types of issues can be made with high accuracy, preventing a significant number of failures that would have notably impacted user experience. Furthermore, Predictive technologies are highly complementary to the existing Reactive technologies that have dominated the Internet for the past four decades.

Moreover, the notion of failures has been expanded to account for issues impacting user experience (referred to as Quality of Experience – QoE). Even if the path might not be entirely down (brown/grey), which was the only trigger for reactive mechanisms, this broadens the scope of failure considerably.

At the time of writing, Predictive Networks have been integrated into several commercial products, with a primary focus on key networking areas like SD-WAN. This technology is anticipated to evolve over the coming years: we expect the emergence of additional telemetry that enables true application feedback-driven predictions and the potential introduction of new algorithms tailored for real-time, short, and long-term predictions. Such technologies will also be expanded to several other networking use cases.

When paired with automation, Predictive Networks could pave the way for long-awaited Self-Healing networks. They have the potential to be one of the most influential technologies for the Internet. Many more innovations are on the horizon.

12 Acknowledgement

I would like to express my heartfelt gratitude to several key contributors with whom I have worked for many years: Gregory Mermoud, PA Savalle, Vinay Kolar, Eduard Schornig. Our collaborative efforts in ML/AI have led to groundbreaking innovations in networking, such as Wireless Anomaly Detection with root-causing, Self-Learning Networks, and detection of spoofing attacks, and Predictive Networks to name a few. Additionally, I would like to acknowledge the work of several highly talented engineers, including Mukund Raghuprasad, Michal Garcarz, Romain Kakko-Chiloff, Petar Stupar, Zacharie Bordard, and Guillaume Nachury, among others, who have made significant contributions to this work. It goes without saying that close collaboration with numerous customers worldwide has been instrumental in achieving such innovative outcomes. I would also like to thank Cisco Systems for their continuous support in bringing new ML/AI technology to life for networking.

13 Bibliography

Cardwell, N., Cheng, Y., Gunn, S., Yeganeh, S., & Jacobson, V. (2016). BBR Congestion Control. ACM Queue*, 20 -- 53.*

Cisco AI Network Analytics*. (n.d.). Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/nb-06-ai-nw-analytics-wp-cte-en.html*

Dasgupta, S., Kolar, V., & Vasseur, J. (Jan 2022). Quantifying Network Path Dynamics using Time-series Features.

SD-WAN: Application-Aware Routing Deployment Guide*. (2020, July 21). Retrieved from https://www.cisco.com/c/en/us/td/docs/solutions/CVD/SDWAN/cisco-sdwan-application-aware-routing-deploy-guide.html*

Vasseur , J., Mermoud, G., Savalle, P., Schornig, E., Schornig, E., Magendie, G., . . . Di Pietro, A. (2023, August). Challenging the Norm: A Groundbreaking Experiment in AI-Driven Quality of Experience (QoE) with Cognitive Networks . *Retrieved from www.jpvasseur.me*

Vasseur, & Vasseur, J. (2023, August). Predictive Networks: Networks that Learn, Predict and Plan - Release v3. *Retrieved from www.jpvasseur.me*

Vasseur, J. (2023, August). AI and Neuroscience: the (incredible) promise of tomorrow. *Retrieved from www.jpvasseur.me*

Vasseur, J., Mermoud, G., Kolar, V., & Schornig, E. (2022). Reactive versus Predictive Networks.

Vasseur, J., Mermoud, G., Magendie, G., & Raghuprasad, M. (2023). Micro Failures, Macro Insights: Unveiling the MIF Phenomenon in the Internet .