SCTE·ISBE CABLE-TEC EXPO 2018 | ATLANTA, GA OCTOBER 22-25

2018 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS

Operational Transformation Via Machine Learning

A Technical Paper prepared for SCTE•ISBE by

Shamil Assylbekov, PhD.
Principle Engineer
Charter Communications
6465 Greenwood Plaza Blvd - Suite 900
720.699.4573
Shamil.assylbekov@charter.com

Devin Levy
Director, Video Operations
Charter Communications
6465 Greenwood Plaza Blvd - Suite 900
3037934199
Devin.p.levy@charter.com

Table of Contents

List of Figures

Abstract

In the current paradigm of Operations acting in a reactive, post hoc manner, something has got to give. MSOs cannot continue to throw bodies at problems in efforts to remediate outages, customer impacting events and impairments. Instead, a data-driven Machine Learning (ML) approach needs to be utilized to change the tide for the better. Cable operators have the ability to merge heterogeneous data from various sources in efforts to qualify and classify problem areas. This will ultimately lead to ML-driven operations which will result in a better sustained customer experience and afford the opportunity for the MSO to work under a lean operations model while employing top engineers to do the work of what would have previously taken dozens of engineers to handle in a post-hoc world. Operational problems go past operations expenditure. Customer impacting events is the name of the game, and by utilizing different ML approaches for different issues, one can classify various events that occurred, provide actionable intelligence and network automation to curtail outages, reduce the mean time to resolution (MTTR) of events in the end providing a better customer experience. Initial efforts of our group have yielded a reduced MTTR for outages and impairments, less escaped defects for feature enhancements, actionable intelligence that affords us the opportunity to make data driven decisions rather than gut reactions or best efforts. This result was mainly achieved via the Deep Learning approach for the purposes of anomaly detection, and a mixture of ML approaches for the purposes of escaped software defect categorizations. At Charter, ML has made Operations a more intelligent organization. Some of the decisions are now made with real time actionable intelligence via our own plethora of analytics; we now have ability to improve the customer experience on a daily basis. And, we encourage the adoption of this new operational transformation, for the betterment of our industry.

Content

Machine Learning has come a long way since the 50s, when it first was conceptualized. In 1952, Arthur Samuel wrote the first learning program to play checkers. The more it played, the better it played. Then in 1997 IBM's deep blue beat Garry Kasparov in game one of chess. Now, we are solving complex multidimensional , operational problems with ML, and in some cases assisting in saving lives too. ML is not just for fun and games anymore, rather ML proves to be a golden opportunity for our industry and our to paths to lessen Service Impacting (SI) events.

In Charter, we use ML to predict and mitigate outages for Spectrum Guide. Spectrum Guide is Charter's flagship video product. It utilizes a plethora of microservices, vendor platforms, internal and external clouds, various databases, and countless application program interfaces (API). Ultimately this is used to deliver an exception experience to the customer where updates and upgrades can be pushed to the box on demand with minimum downtime or interference. With that said though, sometimes things go wrong, and when you have hundreds of different components, you have a lot of opportunity for things to go wrong in a lot of different directions. And that's where ML comes in to play.

One example of ML that Charter uses for Spectrum Guide is for unexpected system crashes. Although one cannot account for the unexpected, one can learn from mistakes. When we first encountered this outage, it took approximately four hours to troubleshoot. Alarms pointed towards areas that were innocuous or ambiguous, and when there are thousands of moving parts, finding the right one as expeditiously as possible is paramount.

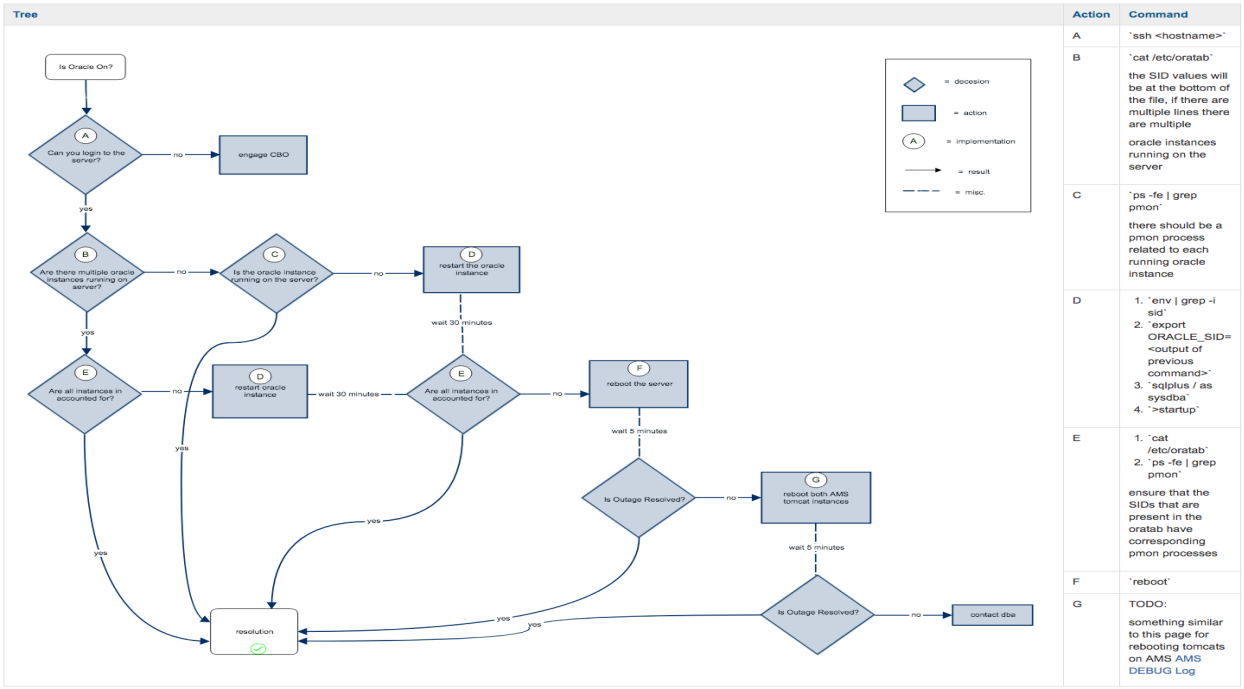

Figure 1 - Decision Tree

When one of the primary the auxiliary databases (DB) experienced a kernel crash at the operating system level, ultimately the database crashed along with it. Any processes that were fed to or pulled from this DB were as a result adversely impacted. After the fault was found, a retrospective took place. We first identified a model for how to identify this type of event, and then put automate in place to follow the model. What we have now is any time this specific issue arises, rather than taking ours to find and resolve we have a reduced MTTR that lasts less than three minutes to resolve instead.

Another example that is more generic but easily adaptable is log analysis. For Spectrum Guide we use time series log monitoring. What we have implemented is anomaly detection. In figure 2 its shown that on August 31st we had a SI event that was immediately followed by automation that reduced the customer impact by 500%.

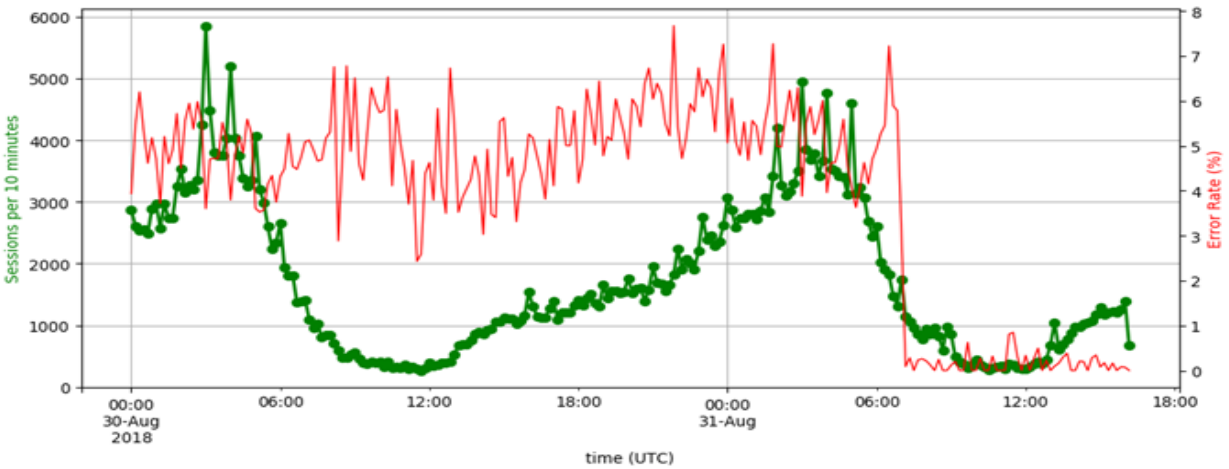

Figure 2 - Actionable intelligence

By training your data either with human interaction or with a neural network, one can achieve quantifiable improvements in the operational space. But ML is not the end all or holy grail, rather it is another tool in our tool chest. It is not just hammer that looks at all problems in need of being hit on the head. Rather, it is only as good as you make it. For the Video Operations Engineering Spectrum team (VOES) , we have had success in several facets of reducing outages and reducing the MTTR of outages. But, there are still many more SI events that we have not trained our algorithms to address. There's always opportunities to make a tree, create binary classifiers, utilize time series plots and ultimately better the customer experience by incorporating ML.

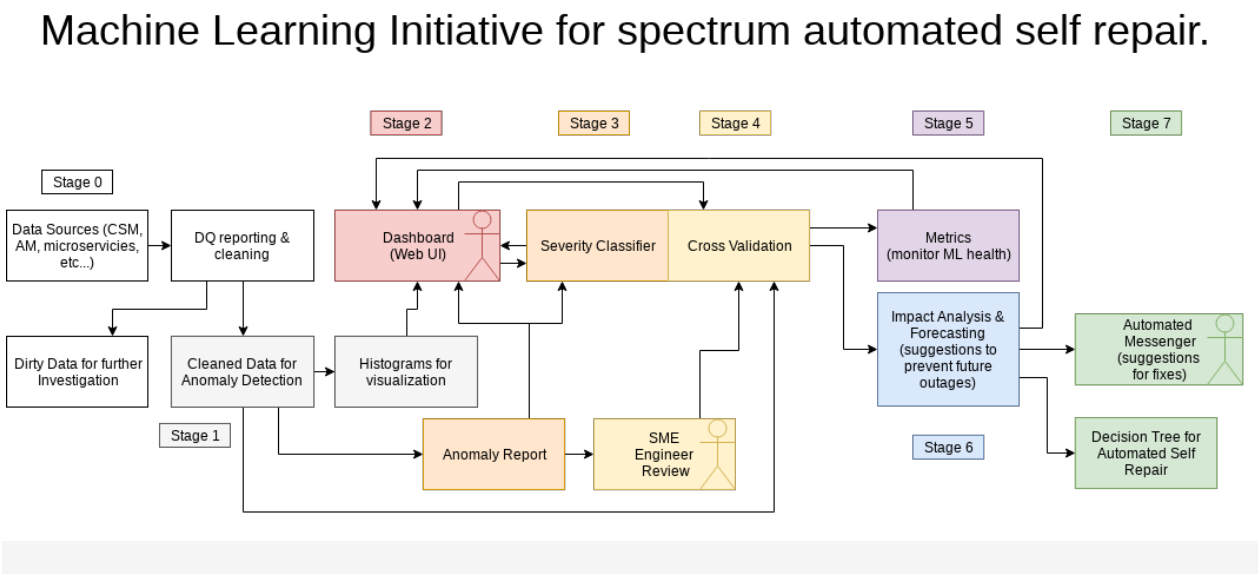

Figure 3 - Process Flow

For more computationally intensive algorithms a distributed computing solution will be needed. Only a subset of one's objectives will be achievable on a small scale and remotely from the actual data files in a batch processing paradigm. One possibility is to set up the clusters or other resources from regional data centers for optimal computational and geographic redundancy. That way the work is done locally to where the data is stored for speed/storage concerns.

In the next phase, one would need to switch investigations from primarily reactive to primarily proactive in order to achieve Anomaly Detection.
This step takes expanded daily reports and generates a meta-report of any anomalous data for investigation. The output of this is a page linked to on the Insights dashboard which lists anomalies with descriptions of the anomaly and relevant images. This will also output a highly curated output with only the anomalous entries for further investigation.

Classification:
As soon as one turns on anomaly detection they will want to classify events based on impact. The meta-report from anomaly detection can then be expanded to include a classification of resulting severity (degree of user impact, outage, this is nothing, engage another team, etc...). Some anomalies are meaningless, while some are indicative of larger problems.

In order to train the severity classification & improve upon it, it would be prudent that the meta-report be reviewed relevant SMEs to see how well the system is assessing the impact of anomalies. The SME

feedback on how meaningful each anomaly is in contributing to solving outages should then be fed back into the severity classifier in order to improve its detection methods.

Note that even after the successful training, machine learning models do not interpret their own results to explain why an anomaly has been flagged. Rather, SMEs will always be necessary not only to train the ML algorithm, but to interpret the needed responses.

Once one has trained on a sufficient dataset of these meta-reports, they can begin operational testing. What is needed at this point is making sure that the classifier is not over-fitting training data by performing k-fold cross validation on our training datasets. The training dataset will be a rolling window of the last X days-worth of data to ensure that the model is based off of, and up to date.

At this point one can consider building/turning on automatic features for detection and different automatic alerts at various levels of severity in an attempt at forecasting problems for repair before they become an outage.

Next, at this point one can begin working on impact analysis to suggest areas for theirs teams to focus in on in order to reduce outages the most efficient way possible. This analysis would look at the anomaly detection from before and attempt to suggest what anomalous features are consistently contributing to calls.

If the ultimate goal is automated repair and self-healing, then the proposed solution is not sufficient. One approach would be to attempt deep learning for automated repair. It could follow a similar paradigm to what is proposed, but would require SMEs to not only flag the importance of an anomaly but list the affected systems AND the appropriate actions taken (read: calling fully tested, generalized, and automated repair scripts) to resolve the problems in those systems. Unsupervised deep learning algorithms would not be wise for automated self-repair.

Any self-repair system should be implemented upon a well understood decision tree providing a classification of the problem where the conditions to entering the automated repair state are a clear and well understood set of comparisons. IF x>someValue AND y<otherValue OR ... Then DoTheThing. The good news is that the classification algorithm described above could be fed into the conditionals of such a decision tree. The classifier shows that the X variable is statistically significantly high and that correlates to outages, so that can be one condition toward calling the automated scripts.

Conclusion

Create a Minimum Viable Product (MVP), it is a good idea and a proof of concept can go a long way.

Picking one data source and implement Insights, Anomaly Detection, and begin building a seed model for Classification. The deep learning feedback loop can be left out at first for the MVP. Object oriented design and proper use of version control will be critical for ensuring long term goals are met while not impairing short medium term goals too much.

Insights and Anomaly detection should prove immediately useful upon completion, and the following stages will need to be trained before their value becomes apparent. If certain details of implementation are handled correctly then the feedback algorithms, meta-report generation, classification, and parts of the anomaly detection should be reusable for other data streams including other sources.

Abbreviations

ML	Machine Learning
MTTR	Mean Time to Resolution
DE	Differential Evolution
NFL	No Free Lunch
SI	Service Impacting
API	Application Program Interface
DB	Database
MVP	Minimum Viable Product