Preventing Network Maintenance Collisions

Using Artificial Intelligence Models for Predicting Collisions in Planned Maintenance Activities

A technical paper prepared for presentation at SCTE TechExpo24

Jordan Kupersmith
Data Scientist II
Cox Communications
jordan.kupersmith@cox.com

Nate Bila
Sr Manager, Technical Project / Program Management
Cox Communications
nate.bila@cox.com

Cherie Peirce, Cox Communications

Chase Durham, Cox Communications

Rob Arnold, Cox Communications

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

1. Introduction

In the digital age, a flawless network isn't about uptime; it's about unforgettable moments—a crystal-clear "I do" in a video wedding, a reassuring voice in a crisis call, or a lightning-fast trade that secures a child's college fund. Yet, as networks converge voice, video, and data services, the manual processes we cling to become silent assassins of these experiences. This paper exposes how current manual approaches to impact analysis and collision detection, when faced with the kaleidoscopic complexity of modern networks, aren't just costly and inefficient—they are actively sabotaging the moments that define customer relationships and brand loyalty.

1.1. When Every Packet Carries a Promise

Today's networks don't just carry data; they deliver life's pivotal moments. The voice clarity conveys empathy in a telemedicine diagnosis. The video fidelity makes a virtual tour feel like being there. The data responsiveness that turns a mobile app into a personal assistant. In this landscape, network quality transcends technology; it's the essence of customer experience (Guo 2021, Gartner 2019). All planned activities must be accounted for and reviewed to ensure that collisions and unintended consequences are avoided. Service providers have systems to catalog and track just about every activity, but how does the knowledge of the activity translate to collision avoidance in an integrated network?

We face unprecedented complexity as we weave voice, video, and data into a single digital tapestry. A video codec's bandwidth appetite might mute a critical voice cue. An innocent data query could freeze a wedding livestream. No longer are these merely technical hiccups—they're emotional fractures and brand trust shattered in real time (Riti 2020, Wu 2020).

Many inputs and considerations need attention when managing change on a network. Current solutions include ticketing systems for the organization of data, dashboards to help group and visualize data, and change lanes, which are predefined days/times for specific changes to occur. These strategies and tools support the organization of planned activities and mitigate risk by assisting the organization of scheduled work by type or internal organization. A change advisory board, consisting of key engineering stakeholders from various disciplines, and boundary partners impacted by the potential change activity such as customer facing teams and network support teams, also promotes collaboration and peer review. Risk is often measured using information such as device type, location of the device in a network, scope of potential impact, and if the activity will or will not affect services. These solutions require human intervention to conduct a complete review and help organize a change activity. Change activities included planned activities supporting network growth and resilience which are managed by various engineering teams.

Companies still rely on manual impact analysis and collision detection in this high-stakes environment. This paper argues that this is not just inefficient; it is a form of organizational malpractice. By dissecting how these manual processes crumble under modern complexity, we reveal their true cost: not in dollars but in disappointed customers, lost opportunities, and eroded brand equity.

2. Manual Impact Analysis: Playing Network Roulette with Customer Trust

A network is designed and connected with physical objects such as modems, routers, and network cabling. The network can be complex, yet despite this complexity, it can be mapped. As the size and scope of the network increases, so does the complexity and effort required to ensure there are no maintenance collisions. When the added complexity of redundancy and multiple paths are included, network reliability can improve, yet identifying collisions can become more difficult. Work can be performed on the physical network, and additional work could be performed on services delivered over the network, which increases the likelihood a collision can occur. These collisions can include reduced bandwidth, degradation of services, or service outages. Having individuals or teams reviewing this type of data can identify collisions, but this process can be tedious and is imperfect.

2.1. False Equivalencies of Manual Analysis in Network Reviews

2.1.1. The False Comfort of Checklists

Network teams often rely on checklists for impact analysis, such as checking voice codec settings, assessing video integrity, and validating data routes. While this method seems thorough, it functions in silos and misses cross-service dynamics. For instance, a seemingly harmless increase in video resolution can degrade voice quality during important calls (Kim 2019).

Artificial Intelligence (AI) would enable change implementers to monitor quality of service (QoS) changes across the enterprise, regardless of technology, ensuring that minor traffic adjustments do not impact voice, video, or data routing by using an anomaly detection algorithm. A common example of anomaly detection is fraud detection for credit cards. The way they work is they sift through millions of records with the assumption that 99.99% are as expected while looking for the 0.01% of transactions that seem to stray from the norm. This could be applied to planned maintenance by cross-referencing unusual network activity with knowledge of maintenance happening on that part of the network. Technicians could be notified in real time to take the appropriate action and the algorithm could learn from these instances in the future to predict what types of planned maintenance events happening at the same time caused unwanted network noise. This can then be applied in the future to know which maintenance events to avoid scheduling at the same time.

2.1.2. The Myth of the Expert's Eye

Relying on individual expertise in converged networks is misguided. Even experienced professionals who have managed VoIP for many years might not foresee the impact of voice changes on video performance. An approved session-initiated protocol (SIP) change, flawless in voice tests, can ruin a CEO's town hall video (Kandula 2008, Handigol 2012).

AI can eliminate human error in checklists by continuously monitoring the network, identifying issues across services, providing predictive analysis from historical data, offering real-time validation of settings and configurations, automating tests, and assessing cross-functional interdependencies, all without the need for human intervention.

2.1.3. Testing in the Lab: A False Sense of Security

Lab tests in controlled environments often fail to predict real-world issues. An update might perform well in a lab but fail under the strain of 10,000 employees joining an unexpected all-hands video call.

AI could enhance testing by incorporating cross-functional information and historical change ticketing data to detect potential issues more effectively (Potharaiu 2015, Xu 2021). AI could also be used to supplement testing processes and include information about cross functionalities and technologies to build a more robust lab environment and detect collisions with services where the lab will only detect collisions with technology. Information from previous deployments, related network services, and historical change ticketing results would strengthen detection.

2.1.4. The Cost: More than Downtime

The consequences of inadequate impact analysis extend beyond mere downtime. A frozen video during a virtual funeral can deny emotional closure, garbled VoIP during a telemedicine diagnosis can shatter patient trust, and data hiccups in a live class can disrupt educational goals. Manual impact analysis not only misses issues but can also lead to these brand destroying scenarios (Marnerides 2018, Accenture 2019).

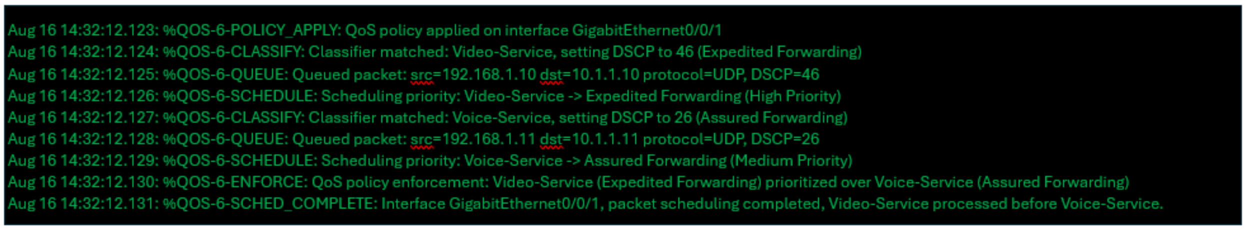
In our current process, for example, QoS changes are viewed as a relatively low risk and can have significant impact on internal and external users.

2.2. Manual Collision Detection: Finding Accidents After They've Already Happened

2.2.1. Log Diving: Aftermath Archeology

"Let's check last week's call quality logs." Great, you've found evidence that video traffic crushed voice QoS. But this isn't forensics—it's customer experience. You're analyzing the aftermath of a week-long customer service nightmare. Those robotic-sounding sales calls didn't just lose deals; they decimated your brand's human touch (Qualtrics 2020, Li 2020).

AI can facilitate the establishment of benchmarks using historical change ticket data and equipment logs. By leveraging predictive modeling and analyzing postmortem data from unsuccessful changes, this enables the development of an accurate impact model based on historical trends from previous changes and their log findings. This approach promotes the ability to look at all change work to determine similarities and risks for impact modeling. This will allow for predictive rather than reactive strategies for identifying potential collisions and assisting in change management before any issues occur.



```
Aug 16 14:32:12.123: %QOS-6-POLICY_APPLY: QoS policy applied on interface GigabitEthernet0/0/1
Aug 16 14:32:12.124: %QOS-6-CLASSIFY: Classifier matched: Video-Service, setting DSCP to 46 (Expedited Forwarding)
Aug 16 14:32:12.125: %QOS-6-QUEUE: Queued packet: src=192.168.1.10 dst=10.1.1.10 protocol=UDP, DSCP=46
Aug 16 14:32:12.126: %QOS-6-SCHEDULE: Scheduling priority: Video-Service -> Expedited Forwarding (High Priority)
Aug 16 14:32:12.127: %QOS-6-CLASSIFY: Classifier matched: Voice-Service, setting DSCP to 26 (Assured Forwarding)
Aug 16 14:32:12.128: %QOS-6-QUEUE: Queued packet: src=192.168.1.11 dst=10.1.1.11 protocol=UDP, DSCP=26
Aug 16 14:32:12.129: %QOS-6-SCHEDULE: Scheduling priority: Voice-Service -> Assured Forwarding (Medium Priority)
Aug 16 14:32:12.130: %QOS-6-ENFORCE: QoS policy enforcement: Video-Service (Expedited Forwarding) prioritized over Voice-Service (Assured Forwarding)
Aug 16 14:32:12.131: %QOS-6-SCHED_COMPLETE: Interface GigabitEthernet0/0/1, packet scheduling completed, Video-Service processed before Voice-Service.
```

Figure 1 – Example of a log

2.2.2. The Great Network Illusion: "It's Up!"

The report from the tools reflects that core routers are 100% up, Voice Over Internet Protocol (VoIP) servers are responding, and video bridges are active. Everything's "up," yet customers are fleeing. Why? The tools see devices, not experiences. They miss that while each component works, their interactions are toxic— QoS settings are making video and voice fight for bandwidth, leaving both mangled (Pelsser 2011, Pan 2010). AI can review the planned activity, and then monitor other outputs that could be affected for impact including bandwidth consumption and peripheral services by scanning monitoring tools in conjunction with the deployment.

2.2.3. Alert Apathy: The Boy Who Cried "Jitter!"

Your system constantly flags potential issues such as voice jitter risk when in fact performance is within standard. This may also appear as video packet loss with a false alarm upon closer inspection, and data congestion incorrectly diagnosed. When a genuine conflict alert appears, it could be ignored, lost in the noise. Result? Your East Coast video launch looks great, but it's silently choking every sales call (Shu 2019, Vaswani 2017)

2.2.4. The Hidden Tax of Manual Methods

There are costs associated with manual methods and imperfect modeling, both to the cable company and the customer. The potential costs to the business are customer churn, labor hours from avoidable trouble calls, and operating expenses from an unnecessary truck roll. The potential cost to the customer could come from an extra charge on a bill due to a technician being sent to their premises or from a hypothetical scenario such as a choppy video interview costing an applicant a job. The candidate may then badmouth the brand for years. Revenue loss from VoIP issues made company reps sound unsure during sales calls, lost sales can multiply globally, and brand erosion is impacted during any event. It is necessary to leverage the best methods through technology to support improved success opportunities. Manual processes can become an operational drain as engineers spend days in war rooms reviewing, planning, and investigating, not innovating. A mental toll is levied on teams, drowning in false alarms, becoming numb and slow to react to real issues.

In converged networks, manual collision detection is not just inefficient—it is an expensive game of whack-a-mole, where every missed issue exacts a compounding, often invisible, business cost (Chen 2015, Wang 2020, Mao 2016)

3. The True Expense: Beyond OPEX and CAPEX

Traditional cost models (OPEX, CAPEX) fail to capture manual management's real price in converged networks. They measure tangibles like labor hours or tool licenses. However, in today's experience economy, the most significant costs are intangible:

1. Lost Lifetime Value: When a customer's video wedding on your network freezes, you don't just lose their $50/month. You lose their family's business forever.

2. Word-of-Mouth Damage: Studies show an angry customer tells 9-15 people. When John's voice garbles during his TED talk, his 10,000 followers hear about it (Silver 2017).

3. Brand Premium Erosion: Companies with top customer experiences command 16% price premiums (Chen 2021). Poor experiences reverse this.

4. Employee Productivity: Video issues in remote meetings don't just frustrate; they measurably reduce collaboration quality (Wang 2014).

5. Innovation Opportunity Cost: Engineers debugging video-voice conflicts aren't creating your next big product.

6. Stock Impact: Major service issues can drop share prices by 3-5% within a day (Rusek 2020).

These costs, invisible in traditional models, reflect a silent need for automation and improved detection. In the convergence era, superior customer experiences command market premiums, while poor ones risk brand value and business viability (Abadi 2016). With a siloed view and slow reactions, manual network management has become an existential risk.

4. Enhancing existing processes with Artificial Intelligence

Since there is already a risk score that is calculated based on the engineers' manual responses to a survey about the planned activity, the next logical step is to enhance it rather than completely replace it at the initial stage. The risk score is a result of simple if/else rules about the activity. For example, if the engineer knows that the maintenance activity will result in customers being offline, or it is a high likelihood, then the risk score is higher. We propose Rule Extraction by Reverse Engineering (RxREN) to use machine learning to further refine the ruleset based on machine learning (Giulia 2020).

This algorithm relies on reverse engineering to prune inputs that are not significant and to discover exactly what drives each aspect of the inputs, even if it is a black box algorithm. It extracts classification rules from the pruned input algorithm in the form of data ranges of inputs learning from misclassified data (Chakraborty 2018).

While this method was specifically developed for neural networks it can be applied to any algorithm, particularly black box algorithms. In fact, the design calls for it to be model agnostic, meaning the internal attributes of the model are not considered. This method is best applied after a predictive model determines the significance of the predictors. In other words, this is an application technique, not a technique to determine the importance of predictors. The rule-extractors must assume that the underlying model is the perfect source of knowledge given a context even if this is not entirely true in practice. RxRen must consider that the outputs of the underlying model to be accurate and should not consider the internal mechanisms of how the underlying model arrives at its output. The results of RxRen must be translatable into a set of if-then rules that can be applied to new potential collisions that are scheduled (Giulia 2020).

We focus on elements with low attribute costs in the pilot phase, which represents the computational effort to get the actual value from the data, i.e. how hard is it to calculate, derive or ascertain the value of the attribute (Giulia 2020). Most of the data is available in the system and it is not too large for computer processing, so data acquisition and processing costs are not what determines attribute costs in planned maintenance data. The attribute costs are primarily determined by complexities of network topology. Information related to network connections, asset relationships, patterns of impact, and text patterns could be leveraged to increase the AI's knowledge and accuracy of identifying collisions.

Network topologies can be complex with many different relationships and interaction points. Considering the design of a network, some assets may be related to a single upstream and downstream asset, while others may have several single downstream assets to consider. Weaving into the mix that additionally some network elements are related to other elements located within another branch to support network redundancy, the physical topology can become large and complex. Another layer that must be considered is the services traveling across the modeled physical path, and which path(s) they are taking for each customer. If physical work is creating a simplex condition, and planned maintenance supporting the service requires a failover, a conflict exists. In many environments, reviews are conducted through team interactions, peer review, and change advisory boards.

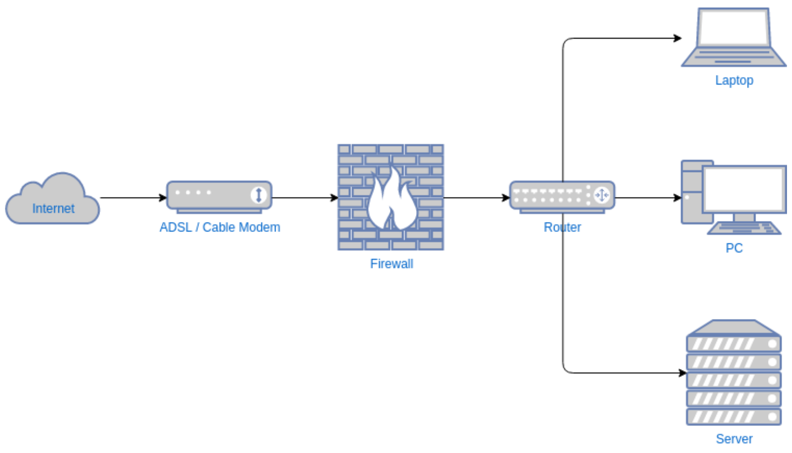


Figure 2 – Example of a Simple Network Topology (Visual Paradigm Online)

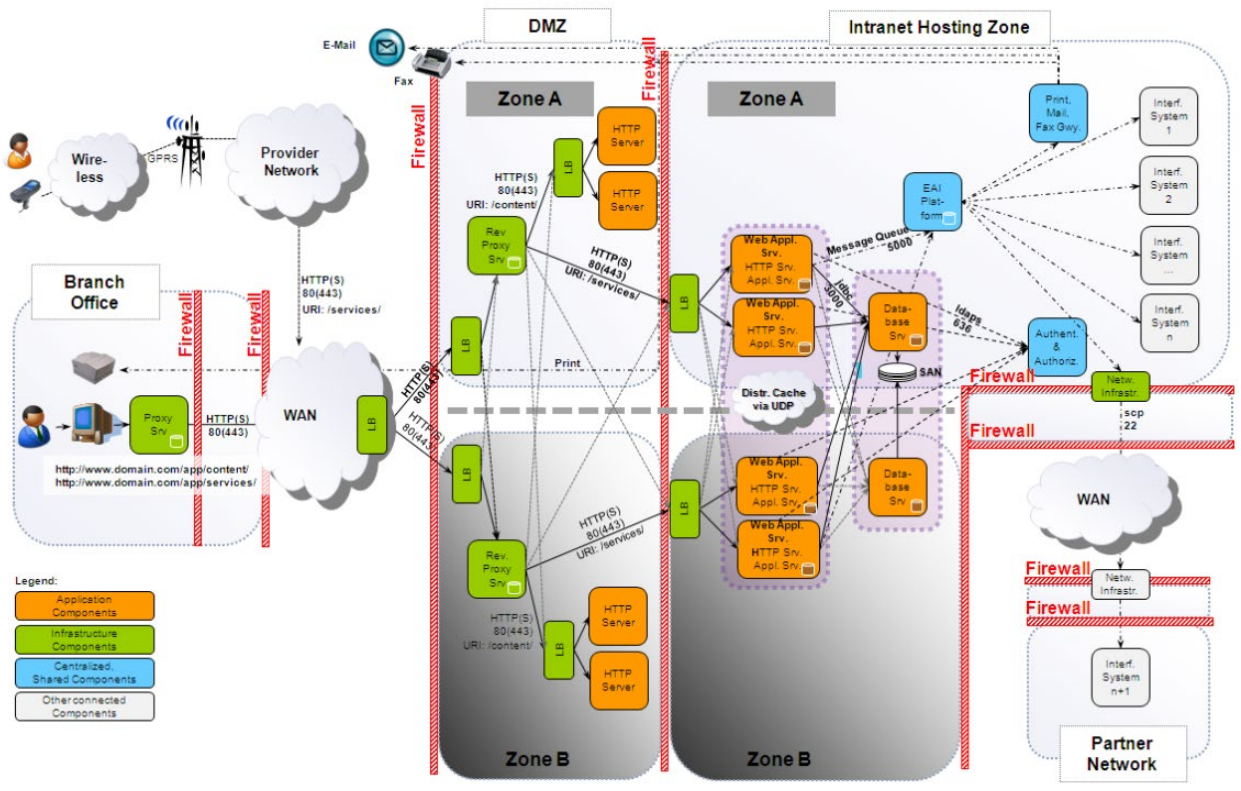


Figure 3 – Example of Complex Irregular Network Topology (Pichler, 2010)

For this reason, we focus on improving the risk scoring system used in the Enterprise Change Request (ECR) ticketing process. The primary objective is to enhance the granularity of risk categories and utilize historical data to predict failures more accurately by using machine learning to analyze past maintenance events and identify patterns associated with failures.

Enhancements to the risk score could be accomplished by incorporating learned data from previous activities such as implementation history, known defects associated with software and hardware, and lifecycle information. In many cases, downstream customer impact can become more clearly defined, enhancing this associated data with specific activity types and locations.

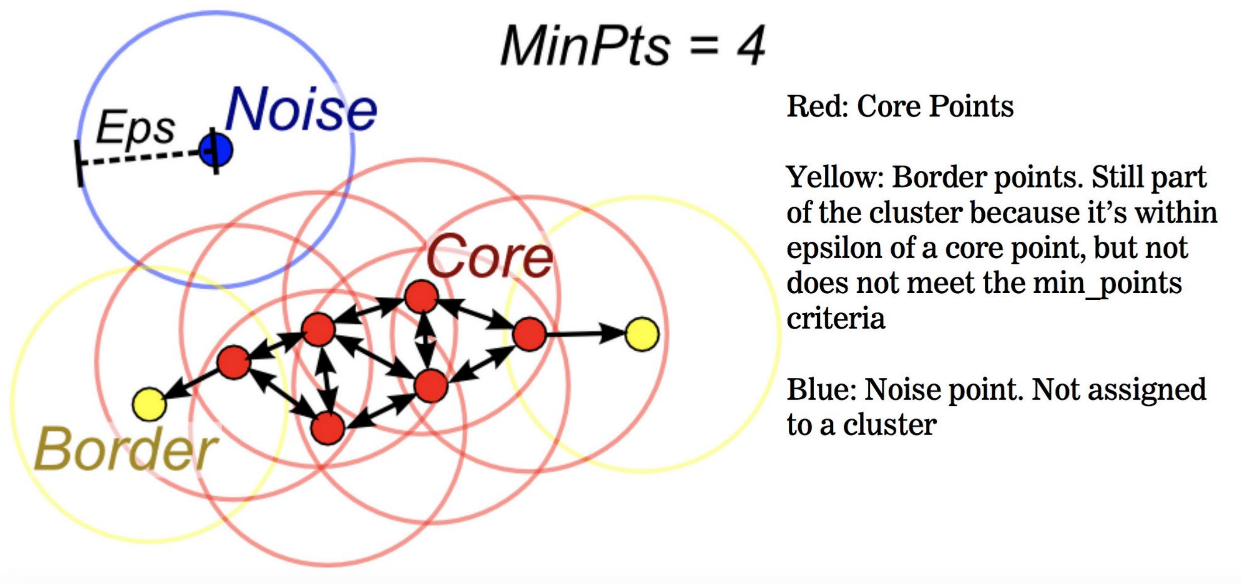
5. Pilot Study

We attempted sorting the dataframe and matching the time overlaps but it was too computationally expensive. The process never finished and was killed after 15 hours. This was expected since pandas dataframes are not quick and the dataframe has 168,927 rows.

We identify duplicates in the dataframe on chassis and date within the same change ticket as a first broad stroke to find collisions, allowing us to decrease our dataset by 50,016 records to 118,911 records. The amount of time it took was not practical to be used in normal operation, so we searched for alternative methods. We successfully implemented the Density-based spatial clustering of applications with noise (DBSCAN) algorithm as a solution due to its speed and its accuracy. It is fast because it only requires a linear number of range queries on the data, meaning a time complexity of O(n). In the worst-case scenario, its time complexity is $O(n^2)$. It also has strength in accuracy because it can detect clusters with arbitrary shapes and the number of clusters do not need to be defined beforehand (Gunawan 2013).

The algorithm works by starting with an arbitrary point and retrieving all nearby points, just like any other clustering algorithm. If the number of points surpasses the minimum points to form a new cluster, then a new cluster is started. It is then expanded until all points that are directly found within epsilon distance, which is user-defined, are found. It then searches for a chain of points that are reachable from these points.

In this example figure from a Medium article by Evan Lutins, we see an illustration of how the algorithm works. Here the red points are high density meaning they core points and therefore in the same cluster. The yellow points are further away from the core points and still qualify as the same cluster, but they are far away enough that if there were enough of them to meet the minimum point threshold then they would be their own cluster. Meanwhile, the blue points are not in any cluster because they are far away from the core and do not have enough points to be their own cluster (Lutins 2020).



Red: Core Points

Yellow: Border points. Still part of the cluster because it’s within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

Figure 4 – DBSCAN Algorithm example

Previous research has established that standard similarity measures like Euclidean distance, as used in k-means clustering, are insufficient for time data and for this reason they is not recommended (Jacques and Preda 2013). One of the reasons is that the k-means cluster algorithm requires setting a predetermined number of clusters, which is difficult for time data. While it would be easy to know the number of clusters beforehand for other use cases like geography where we would know that each data point falls into one of 5 regions in the country, for timestamps over the course of several years we cannot know how many categories there would be beforehand (Tanna 2018).

The parameter tuning required is also easy and applicable to our use case. For example, we know that we need a cluster of more than 1 because if it is a cluster of 1 then there is no collision. Furthermore, since epsilon in this context is merely how long of a time range deviation, we are willing to accept to consider two datapoints to be in the same cluster and from domain knowledge we know that the average maintenance window is 6 hours, it is easy to set.

In our implementation, we set the minimum point to 2 because any two change tickets at the same time is a potential collision and it does not matter if there are only two tickets colliding or more than 2. We set epsilon to 750 minutes or 12.5 hours after a series of trial-and-error experiments led us to determine it to be the best parameter value. The dataset of X was successfully divided into 5,921 clusters. In other words, there are 5,921 times when there were two overlapping planned maintenance events out of a total of 117,538 events. Most importantly, while the sorting method that was previously attempted was abandoned after not finishing after 15 hours, the DBSCAN clustering algorithm took only 1.32 seconds to run.

While 5,921 overlapping events were found, it does not mean that there are 5,921 collisions, just because two events happen at the same time does not mean they are a collision. Events are only a collision if they interfere with each other in space and time, not just time. For this pilot, they are collisions if they are on the same chassis. While the simple string matches are apt for a lookup table of nodes to chassis and then matching each node to a chassis. Then for each combination of cluster and chassis pair, the number of occurrences is counted. If there is more than one occurrence, then it is a collision.

In our dataset spanning from June 15, 2021 to July 22, 2024, we found a total of 1,943 collisions. 1,667, or 86% of the collisions, were only two tickets that collided. However, there were 207 events where 3 tickets collided, 51 events where 4 tickets collided, 5 events where 8 tickets collided, and 10 events where 6 or more tickets collided.

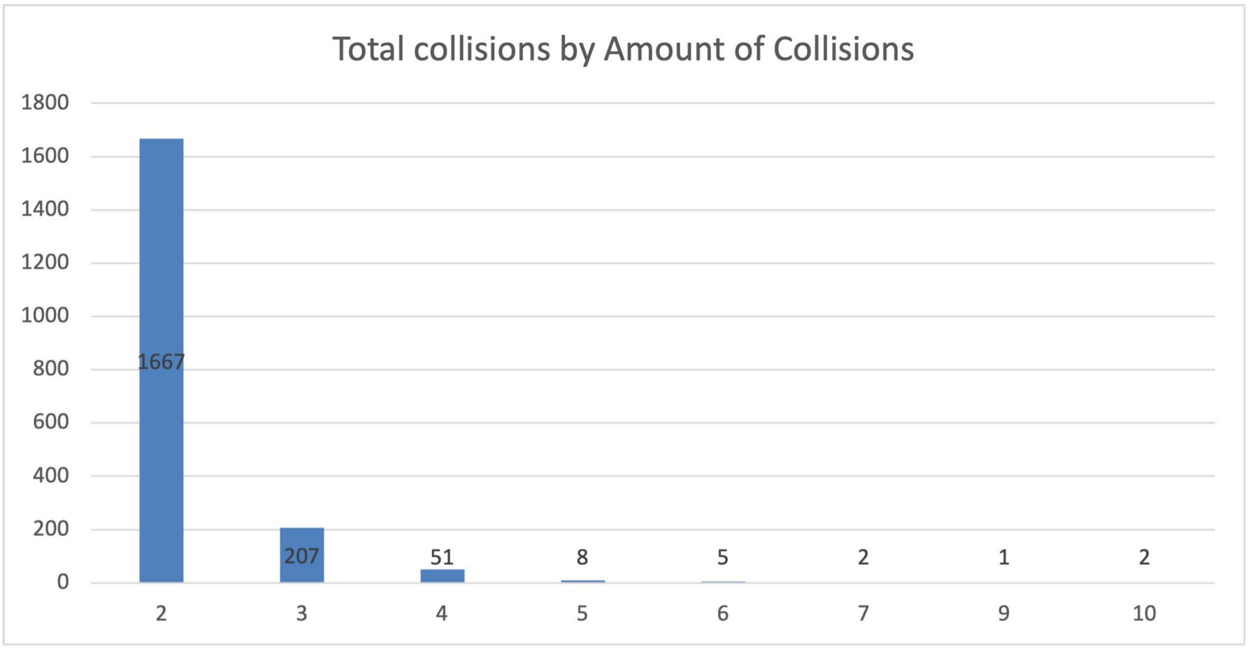


Figure 5 – Total Collisions

While simple string matching is apt for the straightforward example of nodes and chassis, it will not work for more complex elements of the network like identifying when a service traversing a device has work scheduled as the same time the device itself has work scheduled. Examples of a service include applications such as Netflix or VoIP services. Traffic traversing multiple routers while taking its path through the network to reach its destination is difficult to track efficiently. One way to use AI for analyzing services traversing multiple devices in a network is by employing monitoring tools that utilize SNMP or NetFlow, allowing the AI model to analyze the data collected from network device logs.

Having successfully demonstrated in this proof of concept that collision detection is possible, for these more complex collisions in the next stage of this study, we proposed the Bidirectional Encoder Representations from Transformers (BERT) algorithm, which is a new algorithm released by Google in 2018. It is a language representation model that is like the recently famous GPT models. It pre-trains deep bidirectional representations by using both left and right context in all layers as conditions via a masked language model (MLM) with pre-training. This approach contrasts with previous models that only use unidirectional context. In other words, the context is leveraged from both directions.

It is based on the transformer model and uses WordPiece embeddings and serves as a singular model that can be implemented for different tasks, which reduces the need for task specific architectures. The transformer is a mechanism that learns contextual relationships between words or tokens in a text. For every input token in a sequence, a key, value and query vector creates a weighted representation. For our application this is beneficial since there are different layers of the network with different topology configurations. The transformer aspect of the model is key in our decision to use this model because it allows for the ability to understand complex patterns and dependencies within sequences. If our sequences were to have fixed-length patterns, then the simpler Convolutional Neural Network or Seq2Seq model would suffice.

The MLM works by overcoming a previous barrier whereby models were limited to being trained left-to-right or right-to-left because bidirectional conditioning would allow each word to indirectly "see itself" and the model could accidently predict the target word in a multi-layered context. The MLM overcomes this barrier by masking around 15% of the input tokens randomly and then predicts them.

The next part of the algorithm is Next Sentence Prediction which improves the model's ability to understand relationships between sentences or patterns. These relationships go beyond the first-generation Natural Language Processing (NLP) models which only capture words or tokens that are close together. It is this relationship modelling that consumers have grown accustomed to with ChatGPT. It pre-trains for a binarized next sentence prediction task that can be trivially generated from any corpus. Specifically, when using a series of sentences in training, when starting with sentence A as a reference, sometimes sentence B is the next sentence labeled as IsNext and other times it is a random sentence labelled as NotNext. This iterative process is how relationships are mapped. The same logic is how we can determine "how deep" in the network the relationship of text patterns is (Devlin et al 2019).

Perhaps the biggest roadblock in implementing BERT is the fact that in its original implementation it heavily relied on publicly available pre-training data from published books and Wikipedia articles. This does not apply to this context so the biggest challenge is determining the structure of the corpus of our network topology and determining how it can be built by our engineers in an efficient manner. It is possible that this corpus could be built by restructuring a database dump of the combinations of strings that are associated with previous collisions from the postmortem data. The data includes associated events in addition to the event itself.

The benefit of the corpus is that as new collisions or even new potential collisions that are averted develop, they can be added to the corpus. By adding them to the corpus, it makes the model stronger as time passes. Furthermore, using the corpus as the bedrock of model learning, it allows engineers and technicians without any knowledge of data science to become active contributors in the enhancement of the model.

This application of BERT has already been successfully implemented in a parallel application in Information Technology called LogBERT which captures patterns of normal log sequences in online computer systems. The assumption beyond LogBert is that the contextual embedding of each log entry can capture the information of the entire log sequence. This is achieved first through masked log key prediction, which attempts to predict log keys in log sequences that are randomly masked. The algorithm then uses volume of hypershere minimization, which makes normal log sequences close to each in the embedding space. Once the model has been trained, LogBert encodes the information about normal log sequences. This normal log sequence is then used to detect when the pattern does not match and is classified as an anomaly (Guo 2021).

The main goal of LogBert is anomaly detection to enhance the "log diving" process previously discussed in this paper. In practice, this algorithm would learn what normal logs are and produce an error alert when the logs deviate from that norm. This alert could be programmed to be sent to engineers so they can study it and determine if it is a collision or not to be able to better identify collisions in the future. Utilizing generative AI, the possibility exists to limit the human interaction altogether. A model could be built to learn from log files, and historical data to then suggest the best course of action for the engineers to take to resolve the issue.

On top of the BERT models, we then will apply the RxRen algorithm discussed in the previous section. The reason for this two-step dual algorithm approach is due to levels of certainty. The purpose of the BERT algorithm is to harness the power of Artificial Intelligence to find patterns in unexpected places, but these cannot be applied blindly due to potential consequences for error. If we were to allow the algorithm to blindly deny planned maintenance due to collisions, it could end up denying more collisions than it should. While this would prevent the problem of collisions, it would cause another problem by not allowing maintenance to happen when it needs to happen.   For that reason, RxRen will be used to extract rules that the BERT algorithm finds frequently. These rules can then be approved or denied by the change advisory board prior to their implementation. When they are approved, they are used to automatically enhance the risk score, thus allowing more maintenance events to be approved automatically and ideally only flagging events that truly need an expert's attention.

6. Conclusion: Manual Management-A Liability in the Experience Age

This paper has exposed an uncomfortable truth: in today's converged voice, video, and data networks, manual impact analysis and collision detection aren't just inefficient—they're silent assassins of customer experience.

Manual processes, with their siloed checklists and reactive log dives, are artifacts from a simpler time. They see networks as mere conduits, not the delivery mechanisms for life's defining moments. In a world where a video call can carry a marriage proposal or a VoIP line can connect a child with a deployed parent, this technical tunnel vision is not just short-sighted; it is a breach of our customers' trust in our digital services.

The complexity of voice, video, and data convergence in our networks demands a new approach. Manual processes, blind to the rich tapestry of modern digital experiences, have become liabilities. They cost not just money but moments; in today's network world, that's the highest price.

Through the course of this research, it has been proven that manual processes are inherently limited by their inability to respond to the complexities of modern network environments dynamically. These processes, rooted in legacy systems and methods, fail to keep pace with the demands of the digital era, leading to significant operational inefficiencies and increased risk of service outages. The findings underscore that a proactive, automated approach enhances the reliability and stability of network operations and aligns with the evolving needs of customers who expect seamless, uninterrupted service. This shift from reactive to proactive collision detection in planned maintenance is beneficial and necessary for maintaining a competitive advantage in an increasingly saturated market.

However, while the need for automation is clear, the journey from pilot projects to full-scale implementation involves several critical steps that must be carefully planned and executed. In this paper, we demonstrated a simple proof of concept where we found collisions between maintenance events on nodes and chassis. The purpose of this study is to show that collisions that should be easy for a human to detect can go undetected and this was shown to ask what could be missed deeper in the network topology. The next step is to identify more complex collision events and develop targeted pilot programs to assess the feasibility and effectiveness of AI-driven maintenance collision avoidance solutions for them. These pilot programs should be designed to operate within controlled laboratory environments, allowing for rigorous testing of the technology’s capabilities, limitations, and adaptability. The insights gained from these pilots will be invaluable in refining the algorithms, optimizing the system for specific operational contexts, and identifying any unforeseen challenges that may arise during the scaling process.

In addition to the pilot studies, the project requires a comprehensive analysis of the data that it uses. Therefore, an immediate next step involves enhancing the organization's data collection and integration processes. This includes consolidating data from various sources, ensuring data accuracy, and creating a robust data governance framework. By enriching the datasets available for the collision detection AI system, organizations can improve the precision of predictive collision detection models, leading to better decision-making.

Despite the initial investments in Artificial Intelligence (AI) that may appear cost-prohibitive—particularly the need to build a comprehensive corpus of the network topology—the findings of this pilot study demonstrate that it is indeed feasible to start small and gradually expand the scope of AI integration over time. While the multitude of available options can seem overwhelming, our approach shows that incremental implementation is both practical and effective. Additionally, many of the algorithms proposed in this paper are designed to be interdependent, allowing them to complement and enhance each other. As these algorithms evolve to predict collisions in planned maintenance, they can be adapted and refined to address various aspects of network monitoring. This iterative development not only optimizes the AI's utility but also aligns with the broader objective of enhancing the customer experience by ensuring more reliable and responsive network operations.

Abbreviations

AI	Artificial Intelligence
AP	access point
BERT	Bidirectional Encoder Representations from Transformers
CAPEX	Capital Expenditure
Bps	bits per second
CAB	change advisory board
DBSCAN	Density-based spatial clustering of applications with noise
ECR	Enterprise Change Request
FEC	forward error correction
HD	high definition
Hz	hertz
K	kelvin
MLM	Masked Language Model
NLP	Natural Language Processing
OPEX	Operating Expense
QoS	Quality of Service
RxRen	Rule Extraction by Reverse Engineering
SCTE	Society of Cable Telecommunications Engineers
SIP	Session Initiation Protocol
VoIP	Voice Over Internet Protocol

Bibliography & References

Guo, J., et al. "The Evolution of SDN and NFV Toward B5G." IEEE Communications Standards Magazine 5.1 (2021): 18-24.

Gartner. "Gartner Says 70% of Customer Experience Projects Will Use IT." Gartner Press Release, Feb 2019.

Riti, J.S., et al. "Impact Analysis Framework for Voice, Video, Data Convergence." In Proc. NOMS, IEEE, 2020.

Wu, T., et al. "Detecting Network Configuration Conflicts." In Proc. NSDI, USENIX, 2020.

Kim, H., et al. "The Evolution of Network Configuration." In Proc. SIGCOMM, ACM, 2019.

Kandula, S., et al. "What's Going on in My Network?" In Proc. SIGCOMM, ACM, 2008.

Handigol, N., et al. "Mininet." ACM SIGCOMM Computer Communication Review 42.4 (2012): 351-352.

Potharaju, R., et al. "Juggling the Jinx." In Proc. SIGCOMM, ACM, 2015.

Xu, W., et al. "Network Management in the Era of AI." IEEE Network 35.1 (2021): 284-290.

Marnerides, A., et al. "Intelligent Network Management." IEEE Comput. Netw. 134 (2018): 280-301.

Accenture. "Living Services: The Next Wave in the Connected Customer Experience." Accenture Report, 2019.

Qualtrics XM Institute. "2020 Global Consumer Trends." Qualtrics, 2020.

Li, Y., et al. "Learning Attributed Graph Representations for Network Analysis." IEEE TKDE, 2020.

Pelsser, C. et al. "Locating Internet Routing Instabilities." In Proc. SIGCOMM, ACM, 2011.

Pan, S.J., et al. "A Survey on Transfer Learning." IEEE TKDE 22.10 (2010): 1345-1359.

Shu, Y., et al. "Experience-Driven Networking." In Proc. SIGCOMM, ACM, 2019.

Vaswani, A. et al. "Attention Is All You Need." In Proc. NeurIPS, 2017.

Chen, J., et al. "Neural Network-Based Event Detection in Energy Systems." IEEE Trans. Smart Grid 6.4 (2015): 1961-1971.

Wang, Z., et al. "Deep Learning for Video QoE Prediction." IEEE Trans. Image Process. 29 (2020): 8802-8815.

Mao, H., et al. "Resource Management with Deep Reinforcement Learning." In Proc. HotNets, ACM, 2016.

Silver, D., et al. "Mastering the Game of Go without Human Knowledge." Nature 550.7676 (2017): 354-359.

Chen, X., et al. "DeepSpeech: Personalized Speech Enhancement." In Proc. ICASSP, IEEE, 2021.

Wang, J., et al. "Online Feature Selection and Its Applications." IEEE TKDE 26.3 (2014): 698-710.

Rusek, K., et al. "RouteNet." IEEE JSAC 38.10 (2020): 2248-2259.

Abadi, M., et al. "Deep Learning with Differential Privacy." In Proc. CCS, ACM, 2016.

Giulia Vilone, et al. "A Comparative Analysis of Rule-Based, Model-Agnostic Methods for Explainable Artificial Intelligence." AICS, 1 Jan. 2020, pp. 85–96.

Chakraborty, Manomita, et al. "Recursive Rule Extraction from NN Using Reverse Engineering Technique." New Generation Computing, vol. 36, no. 2, 13 Feb. 2018, pp. 119–142, link.springer.com/article/10.1007/s00354-018-0031-9, **https://doi.org/10.1007/s00354-018-0031-9**

Gunawan, Ade. "A Faster Algorithm for DBSCAN" Technische Universiteit Eindhoven Department of Mathematics and Computer Science 2013

Lutins, Evan. "DBSCAN: What Is It? When to Use It? How to Use It." Medium, 4 Dec. 2020, elutins.medium.com/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818.

Jacques, J. & Preda, C. (2013) Functional data clustering: a survey. Advances in data analysis and classification. [Online] 8 (3), 231–255.

Tanna, Vineet. "Time Series Clustering - DBSCAN." Www.linkedin.com, 5 Jan. 2018, www.linkedin.com/pulse/time-series-clustering-dbscan-vineet-tanna/. Accessed 17 June 2024.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., & Google AI Language. (n.d.). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT 2019 (pp. 4171–4186) [Conference-proceeding]. Association for Computational Linguistics. https://aclanthology.org/N19-1423.pdf (Original work published 2019)

Guo, H., Yuan, S., & Wu, X. (2021). LogBERT: Log Anomaly Detection via BERT. Utah State University,. https://doi.org/10.1109/ijcnn52387.2021.9534113

Simple Network Diagram Example. (n.d.). Visual Paradigm Online. https://online.visual-paradigm.com/diagrams/templates/network-diagram/simple-network-diagram-example/

Pichler, M. (2010). *A Practical Guide to Software Architecture*. https://applicationarchitecture.wordpress.com/2010/04/13/diagram-a-more-complex-net-work-diagram-example/

Imgur. (n.d.). *Imgur.com*. Imgur. https://imgur.com/xRwCPoT