Causality for Customer Experience Anomalies with Real-Time vCMTS Telemetry and Machine Learning

A technical paper prepared for presentation at SCTE TechExpo24

Ilana Weinstein
Machine Learning Engineer
Comcast
ilana_weinstein@comcast.com

Ramya Narayanaswamy
Director, Machine Learning
Comcast
ramya_narayanaswamy@comcast.com

Federica Mutti
Machine Learning Engineer
Comcast
federica_mutti@comcast.com

Aaron Tomkins
Machine Learning Engineer
Comcast
aaron_tomkins@comcast.com

Table of Contents

Title **Page Number**

1. Introduction ... 3
2. Background and Related Work ... 3
   2.1. Network Anomaly Detection ... 3
   2.2. Causality and ML Overview ... 4
   2.3. Network Anomalies and Causality ... 5
3. Methodology ... 6
   3.1. Causality Data ... 6
   3.2. Clustering Method and Classification Model ... 7
   3.3. Cluster Interpretation ... 8
4. Implementation ... 11
5. Results and Discussion ... 12
6. Limitations and Future Work ... 13
7. Conclusion ... 14

Abbreviations ... 14

Bibliography & References ... 15

List of Figures

Title **Page Number**

Figure 1 – Network Anomaly and Metadata Example ... 4

Figure 2 – Causal Data Framework and ETL ... 7

Figure 3 – Comparison of Overlapping Clusters with Low Silhouette Score and Well-Separated Clusters with High Silhouette Score ... 8

Figure 4 – Cluster Feature Investigation with Importances and Median Differences ... 9

Figure 5 – Radar Chart Visualization for Clusters Comparison ... 10

Figure 6 – Clustering Workflow ... 11

1. Introduction

Broadband access network continues to experience substantial growth in High-Speed Data (HSD) capacity demands year over year, with customers expecting 100% availability all the time. Internet service is viewed as an essential service, like electric power and gas. All multiple system operators (MSOs) face the daunting challenge of minimizing customer service disruptions and staying ahead of potential issues to detect and mitigate before customers notice the issue.

As we grow our network, Comcast has made significant advancements with Distributed Access Architecture (DAA) by deploying virtual cable modem termination systems (vCMTS). This has enabled a distributed cloud computing architecture, simplifying the deployment of software changes. This architecture allows us to optimize capacity using profile management applications, and deploy new software changes, code upgrades, and necessary updates virtually using cloud computing and Kubernetes pods. On the hardware side, we are constantly innovating on the access layer to support symmetrical gigabit speeds where several hardware upgrades are required, including smart amps, switches, optics, network interface cards, and Remote Physical Device (RPD) hardware.

All software and hardware changes occur on systems with live customers, requiring precise coordination across teams to ensure a positive customer experience. It is also critical to have checks and balances in place to ensure that upgrades to one system do not negatively impact the performance of other systems or applications. This is a complex problem involving a complex system with large volumes of data.

Automated continuous monitoring of our system's health after critical deployments using machine learning (ML) has been covered in a previous paper (Weinstein et al., 2023). To make this actionable, it is imperative not only to detect anomalous behavior but also to identify which system, software, or interplay between systems is causing the anomalous behavior. This is challenging because there is no labeled data to apply supervised learning and systems are constantly being modified, which makes it hard to establish a steady state baseline. Additionally, due to the nature of the systems and components involved, there are many dynamic features, making it difficult to visualize and establish relationships between variables. Automating causation analysis and making it actionable remains a significant challenge.

In this paper we cover an unsupervised learning approach to cluster all relevant features and help in obtaining directionality toward potentially multiple areas that may be contributing to an anomaly. This directionality toward interpretable areas will serve as a good starting point for any manual investigation needed and will aid operations teams and subject matter experts in identifying the source of the problem with the end goal of reducing customer disruption and enabling a positive customer experience.

2. Background and Related Work

2.1. Network Anomaly Detection

The automated continuous network anomaly detection architecture and algorithms are well established and are verified for accuracy. Although there is high accuracy when detecting an anomaly for a single customer experience metric—such as cable modem signal, customer calls, quality of experience, etc.—the algorithms must be aware of the complex external factors that can pinpoint causality. Through the validations, we gathered a plethora of system data required to understand the workings of the network. We can use this data with a causality approach to add visibility to the existing anomalies.

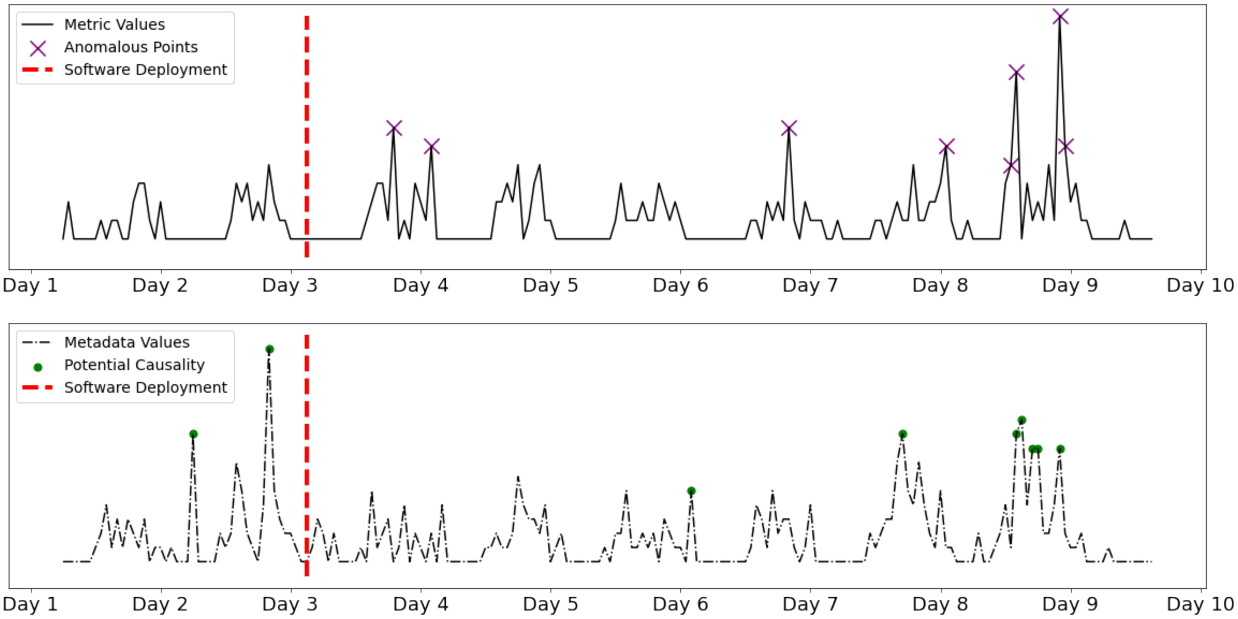

Figure 1 – Network Anomaly and Metadata Example

Figure 1 is an example anomaly that was detected post-software deployment; when just looking at the metric values (top half), it is evident that the anomalous behavior only occurred after the deployment. Looking at this metric alone, one might conclude that the anomaly was due to the software change, but that is untrue. A metadata metric (bottom half), such as node health—a computed score that takes into account Data Over Cable Service Interface Specification (DOCSIS$^{®}$) upstream and downstream metrics—shows that there was some anomalous behavior before the deployment and that the potential causation points correlate to the original anomaly. The knowledge from the metadata lessens the confidence that the original anomaly is change-related and points to a different causation. Using the understanding of network anomalies and their complexities, we aim to develop an automated approach to determine these correlations with ML and make them actionable.

2.2. Causality and ML Overview

A variety of ML approaches exist when it comes to identifying the explanatory drivers or causal factors behind network or system anomalies. Thus, here we will briefly discuss some of the most common approaches to distinguish and highlight how our approach differs and provide some reasoning behind our model choices.

Traditional root cause analysis (RCA) methods such as Pareto analysis, fishbone diagrams, and five-whys (Konstantinos et al., 2022) are very popular due to the benefit of being interpretable and easy to visualize. However, these methods also rely heavily on manual work performed by an individual with strong expertise in all facets of the system and thus tend to be more prevalent when analyzing simpler systems. Due to the dynamic nature of the system in our use case, as well as our desire for automation and minimal manual work, we have moved away from traditional methods as the core of our approach; however, we still leverage some of these methods in ancillary reporting.

Another common approach is based on the topology of the system or network represented as a graph or tree-based structure. Such methods will utilize elementary graph algorithms to identify devices in the topology responsible for causing the anomalies. These approaches are very flexible and can be used in combination with rule-based heuristics and statistical analysis to enable potent monitoring (Simakovic et al., 2021). However, their application also tends more toward situations needing to identify points of complete failure (e.g., devices, nodes) in the graph structure, as opposed to providing explanations outside the topology that might be contributing to issues that are more difficult to detect. Thus, while graph algorithms are used extensively in many of our applications (Lutz et al., 2023), they do not fulfill all of our requirements, particularly for identifying causes outside of network topology.

Causal analysis as a sister branch of statistical analysis is full of a variety of methods, usually meant to quantify the impact (or average treatment effect) of some treatment or intervention of interest. Outside of the experimental design context, one of the most common approaches involves building Bayesian networks as causal graphs from observational data. Such approaches can provide mathematical guarantees for distinguishing associational relationships from true causal relationships and can be the cornerstone in an effective application when combined with subject matter expertise. Bayesian networks can also be set up to mimic network topology but with the added benefit of being able to account for noise in the data and include features and mappings outside of typical network topology (Kandula et al., 2005).

In the context of network anomaly detection, however, an actual treatment effect is not as important as obtaining directionality around the areas contributing to the anomaly. Furthermore, many of the features we are interested in using are such that a causal inference relies more on domain expertise than mathematical proof. Additionally, the directionality toward interpretable areas is also intended to serve as a good starting point for network operators to perform any necessary manual investigations. For these reasons, traditional causal analysis does not form the core of our approach but is instead used in supplementary analysis.

While the above briefly describes some of the most common approaches for finding the drivers behind network issues, they do not alone meet all our requirements, as we explain next.

2.3. Network Anomalies and Causality

For our purposes, directionality that can implicate a potential combination of features (from a comprehensive set) that are driving anomalies is the primary goal. Toward that end, we also want automation in terms of unsupervised relationship discovery and model evolution, as new data and patterns are generated. Finally, we want a general model that makes as few assumptions as possible and that allows much of the problem-specific work to be absorbed in the selection of data features.

Due to our desire to obtain explainability in the context of unsupervised discovery, we have moved toward a latent variable approach that can capture a mixture of explainable features as drivers of an anomaly. One of the most common latent variable approaches for finding underlying drivers of observed phenomena is factor analysis, of which probabilistic principal component analysis (PCA) is a variant. Here, the underlying explanatory drivers are constructed as factor loadings and quantify the amount that each feature contributes to an interpretable explanation. In this approach, a linear generative model is used to infer the underlying drivers that are needed to generate the observed data, and then continuous latent variables are used to associate specific data points to the interpretable explanations (Bishop, 2006).

Since we wanted to allow for the possibility of significant non-linear relationships between features that contribute to an anomaly, we moved toward a compositional model approach that uses discrete latent variables. First, we adopted clustering to discover explanatory drivers in an unsupervised manner. Then, we use these labels in a non-linear model whose output can be understood through a model-agnostic interpretability layer that can help identify the key drivers influencing each cluster.

3. Methodology

This section details the clustering method developed to decrease the time to remediate network disruptions while enhancing dependability through the identification of the issue and prevention of future occurrences. We will discuss the data and preprocessing techniques used, the proposed method, as well as visualizations created to help interpret results for the network operations and engineering teams.

3.1. Causality Data

To effectively explore the causes of anomalies, it is crucial to have an extensive dataset that includes enriched metadata and Key Performance Indicators (KPIs). This dataset serves as the foundation for anomaly detection and understanding causation. The collected metadata covers areas offering insights that assist in pinpointing root causes and is an extension of the supporting data discussed in Weinstein et al. 2023. The main objective of the data collected is to uncover the reason behind each anomaly, link it to a domain in the network, and associate it with a change made to the system or a known cause.

To reach all possible domains of anomaly causality, the causality data encompasses power events, current telemetry events, known activities, device details, and service-affecting vCMTS changes. The metadata gathered includes information about network infrastructure, customer device specifics, and existing telemetry event records. Network infrastructure data sheds light on how network nodes are configured, potentially revealing underlying trends. Device specifics contain details about customer devices to help identify if anomalies are linked to specific device types or customer behavior. Existing telemetry event records provide real-time information on activities like device connection attempts, restarts, and plant health. Lastly, power-related events such as power outages help identify disruptions tied to known issues. Overall, the gathered metadata includes over 30 features.

The supporting data is extracted as a time series and is aggregated to a 24-hour period for each anomaly detected; this is performed to decrease the modeling input size and generalize the activity to the day of the anomaly, as some causations can occur hours before or after the anomaly itself. With the use of metadata and KPIs, we establish a comprehensive framework that can aid in causation investigations and detect if anomalies are linked to devices, services, and/or actions.

Before applying the proposed method, the supporting data is preprocessed to ensure better data quality and avoid misleading results. Specifically, redundant metadata and KPIs are eliminated by applying correlation and association analyses. Additionally, supporting data are scaled or bucketed into broader categories if needed, and metadata that does not provide further information due to being constant or almost constant is detected and removed. See Figure 2 for the causality data workflow and how it is integrated with the proposed method discussed later in Section 3.2.

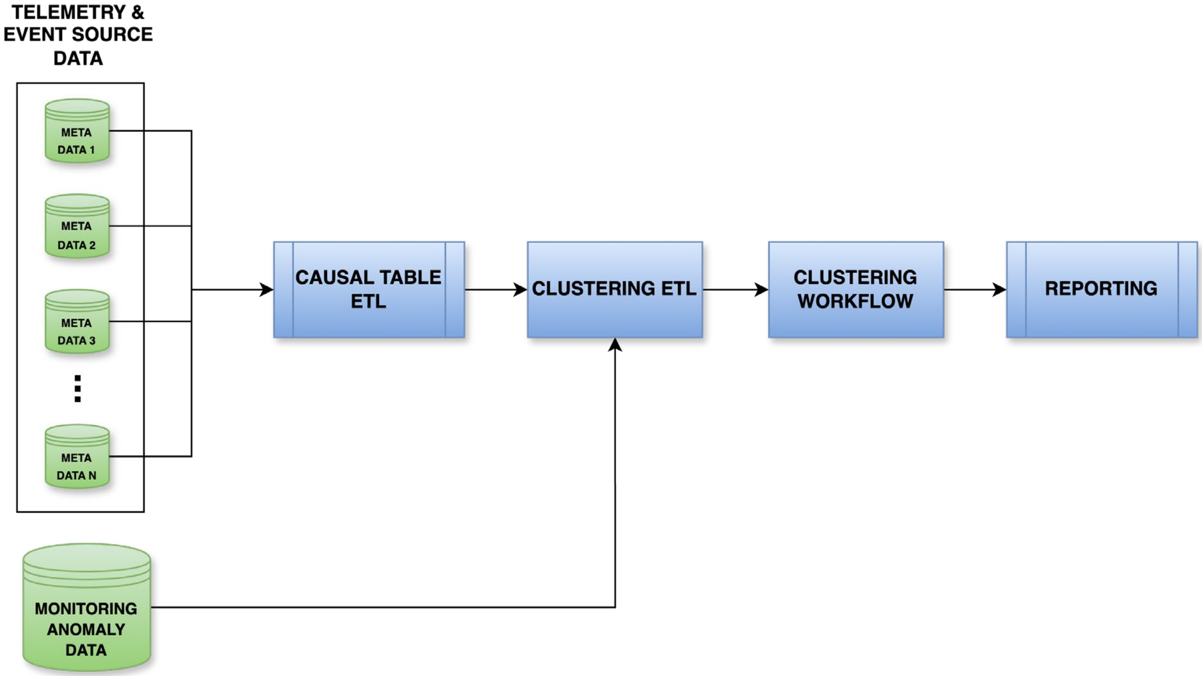

Figure 2 – Causal Data Framework and ETL

3.2. Clustering Method and Classification Model

The proposed approach starts by using a clustering technique to find structural patterns within the data, grouping observations that have similar characteristics and behaviors. Clustering can be compared to organizing a vast library of books not just by genres but also by specific attributes such as frequency of use, popularity, author, and year of publication.

In the exploration of network anomalies, the goal of partitioning anomalies into various clusters is to identify sets with shared metadata and/or KPIs. This strategy is the first step towards uncovering the root causes of anomalies, as it detects groups that show similarities, thus simplifying the identification of factors that may lead to the detected anomalous behavior.

The clustering method used here is a K-Prototype algorithm, which can handle mixed data types. It calculates the Euclidean distance for numerical variables, mirroring the K-Means method, and measures the dissimilarity for categorical variables, like K-Modes. Then, it utilizes a parameter to balance the influence of numerical versus categorical variables in cluster assignment. This hybrid methodology makes K-Prototypes well-suited for analyzing the given diverse causality data in the ever-evolving network, managing the complexity of different data types.

To improve the performance of the clustering method, an exhaustive grid search was performed to optimize the selection of multiple pre-processing and clustering parameters. This experiment explored nearly 600 combinations of parameters, such as correlation thresholds, association thresholds, scaling types, and methods for managing categorical features and detecting constant/quasi-constant features. Each parameter combination was assessed using the Silhouette score, a metric that evaluates the quality of clustering by measuring each data point's similarity to its cluster and its separation from other clusters. By averaging the Silhouette scores across all data points, an overall Silhouette score is obtained, providing a comprehensive measure of the clustering quality. Figure 3 provides examples of both overlapping and well-separated clusters, along with their corresponding Silhouette scores.

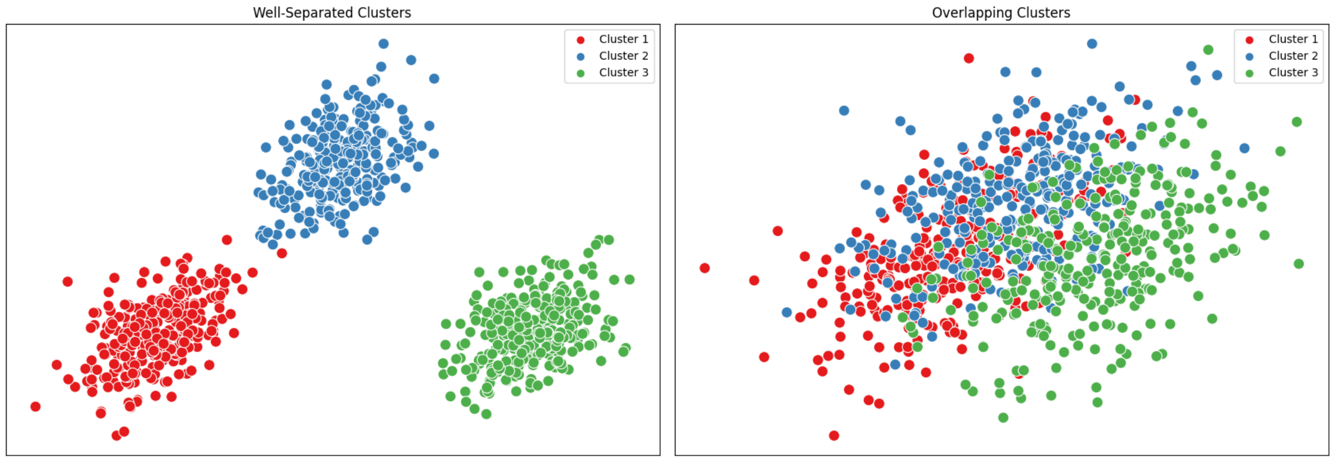

Figure 3 – Comparison of Overlapping Clusters with Low Silhouette Score and Well-Separated Clusters with High Silhouette Score

The combination of parameters yielding the highest silhouette score was chosen for implementation of the proposed approach. This optimal choice ensures the most effective pre-processing for the given network data and fine-tunes the clustering algorithm to achieve the best possible results.

Once anomalies are grouped into several clusters, the next step to enhance explain ability involves adding an ML layer. In this layer, a tree-based classification model is trained for each identified cluster. Specifically, for a given cluster i:

$$Y_i = 1 \text{ for anomalies within cluster } i$$
$$Y_i = 0 \text{ for anomalies in all other clusters}$$

This binary classification setup enables the predictive model M_i to identify the key factors that distinguish anomalies in cluster i from those in the remaining clusters. The performance of the model is evaluated using common metrics such as the area under the curve (AUC) and the F1 score, ensuring that the model achieves high precision and recall in detecting anomalies within cluster i.

In the final step, to further enhance explainability, Shapley Additive Explanations (SHAP) values are calculated for each predictive model to enhance interpretability. The idea behind SHAP is to assign an importance value, known as the Shapley value, to each feature (network metadata and KPIs in the given data) used to train and test the model. These SHAP values quantify the impact of each feature on the model’s predictions. By computing these values, it is possible to identify the most important features that influence individual anomalies and anomalies within a specific cluster i. This last step helps uncover the key factors and underlying causes of these anomalies, further facilitating the explanation of causality for network anomalies.

3.3. Cluster Interpretation

Interpreting the clustered anomalies helps deliver the clustering results to engineers and network operators and as an internal check of the algorithm's performance. All visuals and interpretations are used together to help determine causality for anomalies. A granular view of the individual anomalies is also needed to drill down into the clustering results.

The first visual is a 2-dimensional plot of the clusters generated. Although we use the Silhouette score discussed in Section 3.2, the 2-dimensional visual is also a helpful gauge of performance and provides a deeper understanding of clustering to Subject Matter Experts (SMEs). See Figure 3 for how this could look; the more distinct the clusters, the more accurate the model separates similar anomalies. It also can indicate the nature of the underlying anomalies and if it is possible to cluster based on the data set; if it needs to be more obvious where to separate anomalies based on the visual, it's just as hard for the machine to do.

The following interpretation helps the understanding of the reasons for the divisions between each cluster. It uses the additional ML tree-based classification layer, the resulting SHAP values, and the raw anomaly metadata values to provide causality for each cluster. The first half of the visual is a feature SHAP value comparison across each cluster, indicating which feature is most important in creating the cluster based on the classifier. The next half compares the median value for each feature as a cluster to the rest of the clusters. The two sections are used together to determine the following: (1) which feature is most important to a cluster, (2) what the value of that feature is, and how much it differs from the other clusters. If a feature is only necessary in one cluster and has a value in a critical direction for that feature, it can be deemed the anomaly's cause. Like the example in Figure 4, if the first cluster’s most important feature is the number of customers with a power outage, the average value is that there were outages, and no other cluster shares that importance, then that is the cause of the group of anomalies.

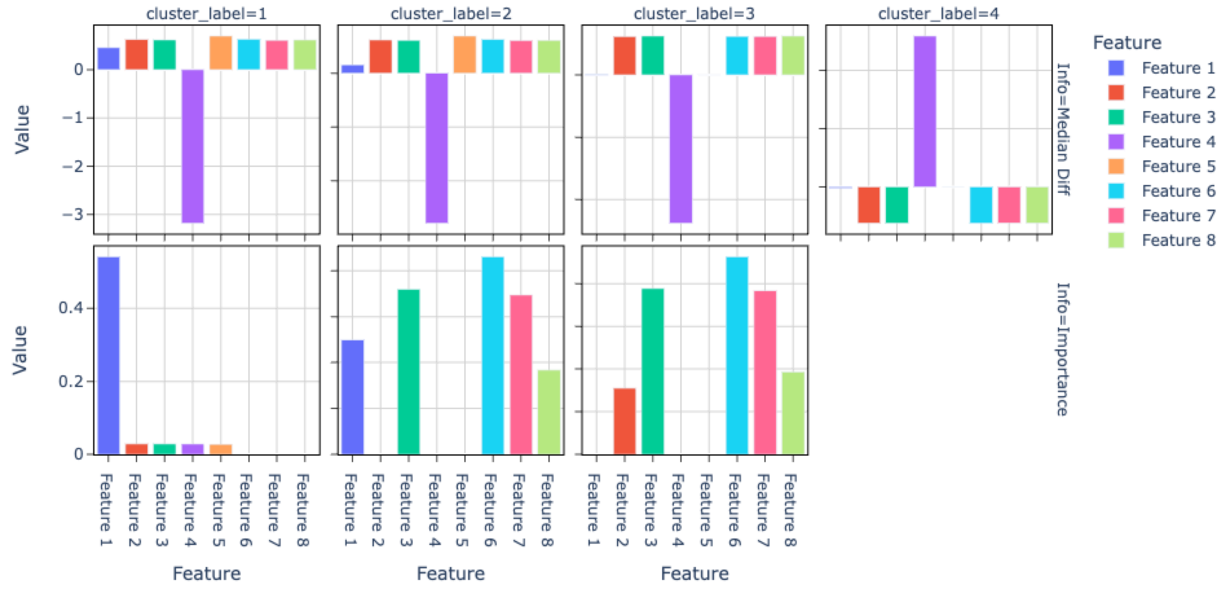

Figure 4 – Cluster Feature Investigation with Importances and Median Differences

Another interpretation provides statistics for clusters tied in information layers that are useful to network operators. This interpretation breaks out each cluster and adds another layer with corresponding summary statistics. Some possible layers are the explainable (known activity, existing event, etc.) vs. non-explainable, DAA deployment, and devices. The supporting statistics include the average number of days since the last deployment the anomaly occurred, the number of times that anomaly repeated, and the percent of a cluster this breakdown falls into. Based on SME input, the closer to the deployment, the anomaly occurs, and the larger the number of repeats, the higher the priority of the anomaly. The user can use this breakdown of anomalies to help prioritize anomaly investigation. Suppose the percent of anomalies in a cluster belonging to these layers is high, and other statistics are alarming to the SME based on domain knowledge, in that case, the anomalies in the cluster should be investigated.

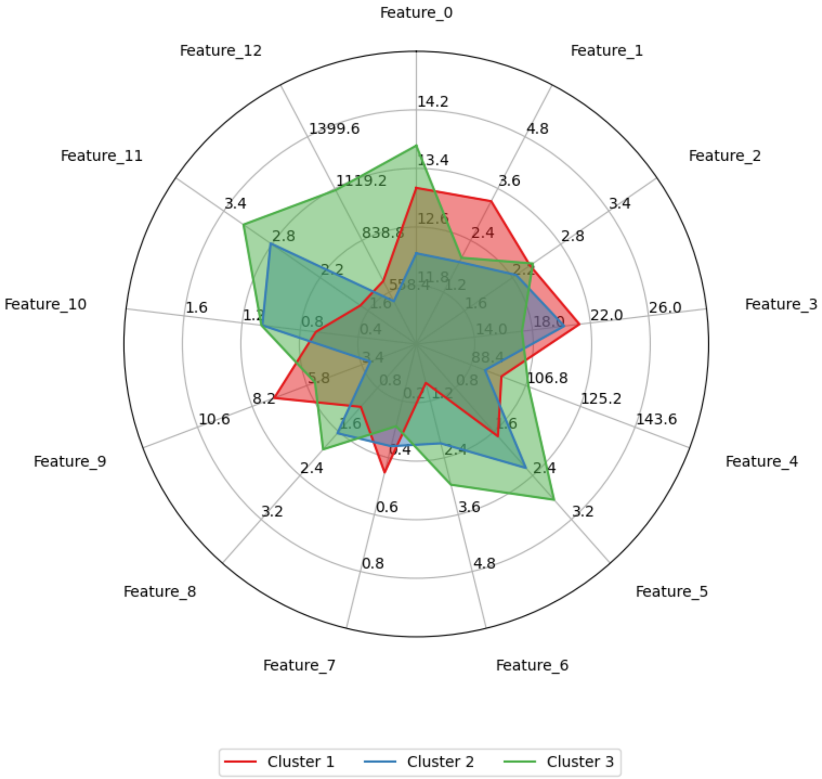

Figure 5 – Radar Chart Visualization for Clusters Comparison

Figure 5 exemplifies a radar chart visualization, which facilitates the comparison of clusters based on their average feature values. This is a further effective tool for summarizing clustering results and supporting the understanding of how specific features impact each cluster. Specifically, each axis of the chart corresponds to a different feature and the values plotted along each axis represent the average value of that feature within a given cluster. For example, Cluster 3 (shown in green) has higher average values in features 0, 5, 6, 8, 11, and 12 compared to other clusters, clearly delineating the predominant attributes of this group.

The last interpretation is a granular view of each anomaly and bucketing them into their respective domain. Along with the SHAP values for the overall cluster feature importance, these values are also available at an anomaly level. The anomalies, along with their metadata, can be split up into the feature domains discussed in Section 3.1, using the most essential feature (learned in the classification layer) to aid in diving into all anomalies in the clustering data set. The method provides further guidance on potential causal relations of the anomalies.

Overall, these interpretations allow for clear communication and enhanced understanding of anomalies and clusters among stakeholders. The discussed visuals can also highlight causality, allowing network operators to quickly pinpoint causality without sorting through all individual anomalies. All supporting visuals are also useful for internal review, keeping a pulse on the algorithm's performance and results.

4. Implementation

A workflow solution has been developed to implement the clustering and classification approach, aimed at exploring causality in network anomalies. This system then summarizes the findings and shares them to relevant stakeholders on a predefined basis.

The workflow works on a powerful analytics platform that has big data processing capabilities and ML tools. The choice to develop and deploy the causality solution on this platform was influenced by several reasons:

- The platform also supports the anomaly detection solution, providing centralization of tools and processes.
- Designed for collaboration, the platform facilitates sharing and contributions across the data science team.
- It makes it easy to interact with the system, deep-dive and perform analyses on the results, as well troubleshoot updates.
- The platform has advanced workflow and scheduling capabilities, essential for automating and scaling the causality system.

Each run of the workflow begins with an extract, transform, load (ETL) phase, where network metadata and KPIs are collected for anomalies detected in the past X days. The data is then processed in preparation for clustering. During the clustering phase, the optimal number of clusters k is dynamically determined: a range of values for k is tested by performing clustering on the current data, and the k value with the highest Silhouette score is selected for that specific run.

Following the clustering, the ML phase is executed, generating the feature importance information for each cluster. The outputs of this phase, along with clustering results (metadata, KPIs, and the cluster each analyzed anomaly belongs to), are saved into an ML tracking and data storage system at the end of each run.

A summary of the run’s output is then created to present results in a clear and concise format, providing information that aids in the prioritization of groups of anomalies for investigation. The details of this summary output are discussed in Section 3.3. In this final phase, the causality report is automatically shared with stakeholders through a cloud-based collaboration platform to facilitate communication.

This structured architecture not only ensures that the system remains flexible to changes in the data, such as the inclusion of new network metadata, but also maintains the highest standards of data preprocessing and performance. See Figure 6 for an overview of the clustering workflow, a detailed view of the causality workflow seen in Figure 2.

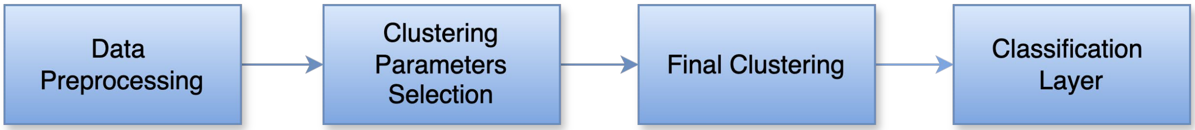

Figure 6 – Clustering Workflow

Regarding the operational specifics:

- The workflow setup includes a driver with a moderate number of vCPUs and a scalable number of workers, each with a higher vCPU count, providing substantial computational power to handle varying workloads.
- Each run of the workflow can take an hour or more, depending on the number of anomalies detected in the past X days. The greater the number of anomalies detected, the longer the run will take to complete.

5. Results and Discussion

Using clustering on detected anomalies to define causality, we encountered crucial patterns and insights that help understand this method. Here, we delve into these discoveries and what they mean for the methodology and practical use.

Our investigation has shown that clusters typically remain consistent, with 3-4 clusters detected in each run. However, when new abnormal patterns emerge with new causations, both the quantity and quality of these clusters can change. This dynamic aspect of clustering highlights the importance of techniques that can adapt to patterns as they appear, ensuring the effectiveness and relevance of anomaly detection.

A common trend in our method is the presence of a "catch-all" cluster that groups many anomalies, while the remaining anomalies are distributed across 2-3 smaller clusters. One primary factor driving one of these clusters is the node score for anomalies detected during a maintenance window, which likely indicates plant issues as the main root cause. Another frequent cluster is related to the customer premise equipment (CPE) firmware version, pointing to incompatibilities observed between the software and certain CPE firmware versions. This issue is usually hard to detect and diagnose as root cause, but the proposed approach can identify and address it more effectively. The last cluster, if it appears, is more dynamic and driven by new emerging abnormal patterns.

This over-arching clustering trend suggests that including additional metadata could improve the breakdown of anomalies into smaller and more precise clusters. By presenting the anomalies within the clusters to SMEs, we can discover more features to include. This ongoing process of tuning the input data demonstrates the importance of comprehensive data in enhancing the accuracy and usefulness of clustering algorithms.

Another discussion point in the current approach is that some metadata showed low variation or strong correlations during feature selection, excluding them from the clustering process. This has prompted us to begin exploring techniques that can handle these types of attributes without penalizing them for being interconnected: either addressed in pre-processing or the causality method as part of future work (Section 6). Addressing this challenge will strengthen the reliability of our model, ensuring that the causal analysis model accounts for valuable data.

Visualization techniques have validated the discussed clustering approach. These visual aids effectively categorize anomaly types and offer insights into what causes them. However, attributing cause and effect based on clustering poses challenges, indicating a need to explore enhancing these visual and analytical methods to draw more definitive causal connections.

While clustering proves effective for grouping similar anomalies, it also demands computational resources, mainly when applied to a large dataset. This insight is crucial for designing and expanding anomaly detection systems in settings with resources. Future iterations of the model will have to find a balance between the thoroughness of analysis and the availability of resources by using efficient algorithms and implementations.

The practical implications of our clustering approach are not only significant but also highly relevant. By integrating analysis with clustering, we can streamline the process of addressing anomalies, a precious asset in settings where a multitude of anomalies surface daily. Prioritizing anomalies by considering their causes and utilizing domain knowledge helps reduce customer impact and effectively allocate resources to address the issues. The clustering approach has been in use for some time now and has been useful in prioritizing the investigation of anomalies with device-specific issues. Since these anomalies have the same metric signature as others but slightly different metadata, they could have gone under the radar if not for our approach.

Our current implementation of anomaly detection and clustering has the flexibility to add additional KPIs and metadata for which we can track anomalies and add to causation models respectively. The current implementation can easily be extended to monitor the health of the system close to real-time and provide causation on anomalies observed, which helps network operators immensely.

Furthermore, our approach is very general and does not depend on the specifics of a given network to be embedded into the model architecture, instead allowing feature selection to absorb many of the system specifics. For this reason, our approach can easily be set up and applied to many different networks.

In conclusion, while our current approach shows promise, the insights and challenges we've discussed present exciting opportunities for advancement. By addressing these areas, we can refine our causality methods to be more precise, efficient, and beneficial in real-world scenarios.

6. Limitations and Future Work

The proposed clustering method combined with an ML layer is a powerful foundational tool that meets our goal of automated anomaly detection and causality. Although the current approach is a good step in the right direction, it has several limitations.

Firstly, the metadata and KPIs related to a given anomaly are aggregated at a 24-hour interval. Aggregating data daily, rather than relating it precisely to the time of the anomaly, can result in a loss of information. Also, as discussed in Section 3.2, the proposed methodology involves a two-step modeling process: a clustering method followed by a classification model. This dual-layered structure decreases interpretability and adds complexity, which may impact the overall performance and accuracy of the approach.

One potential way to overcome both limitations is to maintain the time series aspect of this problem and develop a classification model where the metadata and KPIs at the time of the anomaly are used as input space, with the target to predict whether the observation is an anomaly or not. This single-layer approach, combined with the use of SHAP values, will help distinguish the most important factors causing an anomaly at a specific time t versus other non-anomalous observations recorded at different times.

An alternative approach is to use a continuous latent variable model as mentioned in Section 2.3. This approach will provide a matrix of factor loadings that can shed light on hidden causes behind network anomalies. Such a model will also allow for a time series dependency structure, with distributions computed through the Kalman filter and parameters estimated using Bayesian sampling or variational inference. Non-linearity could be achieved with a variational autoencoder-type architecture, in which case we could continue to utilize SHAP for explanatory purposes.

Another aspect of future work focuses on addressing Physical Point of Deployment (PPOD)-level anomalies, as the current approach is only applied at the RPD level. Also, there is an interest in moving towards a full footprint of anomalies, not just those PPODs that have recently undergone deployment. This expansion will enhance the applicability of the approach across different network levels, ensuring a comprehensive understanding of anomalies. Our approach can easily scale to full footprint if we limit the window of monitoring data that we process simultaneously. Additionally, while there is not necessarily one standard way to parallelize our chosen clustering method, our architecture is not dependent on any specific clustering algorithm or fitting procedure (e.g., batch, iterative, online). Due to the flexibility of our approach, for example, we could substitute another model into the clustering module (e.g., Gaussian mixture, probabilistic PCA) whose fitting procedure scales better with any arbitrarily large dataset.

7. Conclusion

This paper introduces an unsupervised way for network anomaly causation that helps identify potential key factors behind customer experience anomalies. By utilizing K-Prototype clustering, SHAP values, and machine learning techniques, we grouped similar anomalies and provided actionable insights that the network operations team can start troubleshooting to limit customer impact.

Our current implementation allows us to continuously monitor our systems, not just during deployments but around the clock. This enables us to detect anomalies ranging from those causing large-scale customer impact to those affecting only a few customers, which might go unnoticed by traditional tools. By identifying features that help explain the causes of these anomalies, our data-driven, automated approach manages a complex system that successfully identifies anomalies and causalities to investigate, minimizing customer impact. Future work will include new data processing techniques, multi-level analysis, and full-scale models. Overall, this framework ultimately helps us achieve the goal of improving the customer experience.

Abbreviations

AUC	area under the curve
CPE	customer premise equipment
DAA	Distributed Access Architecture
DOCSIS	Data-over-cable Service Interface Specification
ETL	extract, transform, load
HSD	high speed data
KPI	key performance indicator
ML	machine learning
MSO	multiple system operator
PCA	principal component analysis
PPOD	physical point of deployment
RCA	root cause analysis
RPD	remote physical device
SHAP	Shapley additive explanations
SME	subject matter expert
vCMTS	Virtual Cable Modem Termination System

Bibliography & References

Bishop, Christopher. Pattern Recognition and Machine Learning. Springer Science and Business Media, LLC. 2006.

Kandula, Srikanth; Katabi, Dina; Vasseur, Jean-Philippe. Shrink: A Tool for Failure Diagnosis in IP Networks. 2005.

Lutz, B., et al. (2023). Graph Algorithms and Real-Time Telemetry for Intelligent Plant Operations. Paper presented at SCTE Expo 2023.

Papageorgiou, Konstantinos; Theodosiou, Theodosios; Rapti, Aikaterini; et. al. A Systematic Review on Machine Learning Methods for Root Cause Analysis Towards Zero-Defect Manufacturing. 2022. https://doi.org/10.3389/fmtec.2022.972712.

Simakovic, M.; Cica, Z. Detection and Localization of Failures in Hybrid Fiber-Coaxial Network Using Big Data Platform. Electronics 2021, 10, 2906. https://doi.org/10.3390/electronics10232906.

Weinstein, I., et al. (2023). Scaling DAA: Smart, Continuous Network Health Monitoring for vCMTS with Machine Learning. Paper presented at SCTE Expo 2023.