Prodapt

Predictive Network Fault Detection

ML-based proactive approach to spot & address network fault ahead of occurrence

Credits
Rajguru R; Venkatachalam Ramakrishnan; Harish G

Current state of **Network Operations Center (NOC)** in the connectedness industry

The growth in 5G, IoT, and other advanced technologies is increasing network complexities and the demand for real-time monitoring and rapid issue resolution

Network virtualization and SDN are forcing NOCs to manage virtual functions rather than just overseeing physical infrastructure

According to Ericsson's report predicts 1.5 billion 5G subscriptions and 4.1 billion IoT connections by 2024, potentially overwhelming incident management capacity. Managing a significant influx of alarms may lead to delays and strain Network Operations Centers (NOCs), impacting response efficiency

Navigating Obstacles in Network Operations: Predictive alarm classification and ticket prioritization

Network Operations Centers (NOC) are grappling with the challenge to mark out critical alarms from a variety of often irrelevant ones. A robust machine learning (ML) model to streamline issue prediction and resolution within NOC operations can be a potential remedy.

Major challenges faced by Service Providers

- Unprioritized alarm triggers and tedious manual reduction of their volume through rule-based methods
- False alarms inflate costs, causing network delays and unwarranted expenditures on service visit
- Lack of self-healing for alarms underscores the need to pinpoint redundant or non-urgent alerts
- Complex networks hinder precise alarm root cause identification

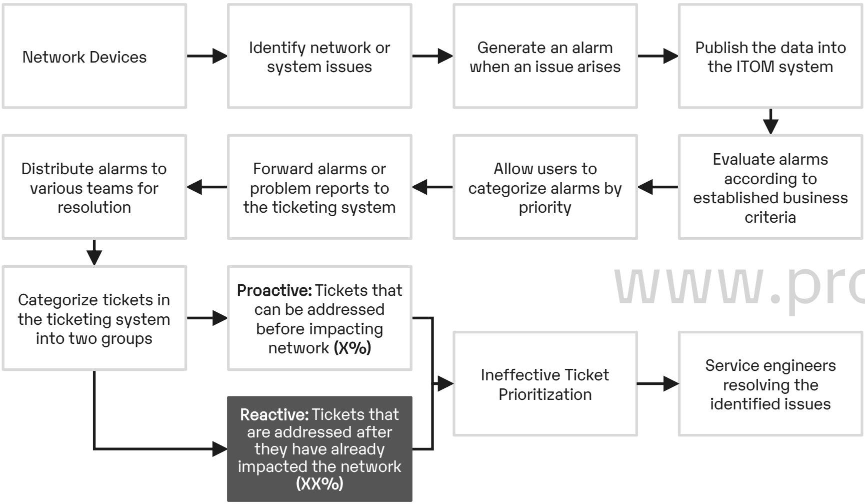
Impact of challenges/pain points

- Network Downtime Issue/Loss of Customers
- Service Outage/Cost of Poor Quality of Service
- Compliance Issues
  For example:
  - Fire Accidents in Devices
  - Problems Specific to X.733 Standards (ITU has crafted standards specifically for the precise management of telecommunications networks.)

> To overcome these challenges, service providers must shift from manual or rule-based reactive network fault detection towards a proactive approach, utilizing ML algorithm to predict network faults.

Orchestrate predictive network fault detection models for the NOC

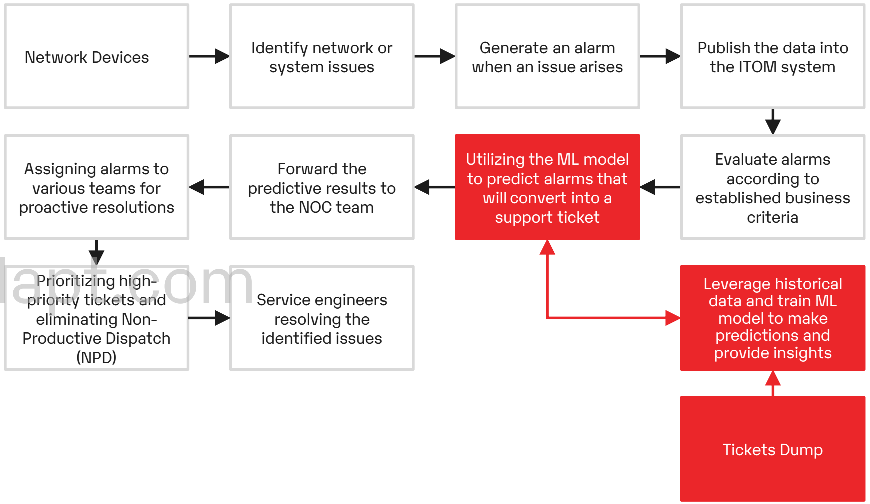
Ticket prioritization challenge during multiple alarm triggers



The **predominance of larger highlighted reactive tickets over smaller proactive** ones leads to more frequent service disruptions, diminished network reliability, and an increased risk of customer dissatisfaction due to delayed problem resolution and heightened network challenges.

Network complexity	Reactive Approach of Issue resolution	Ineffective ticket prioritization

Proactive ticket prediction & prioritization model



Key areas for ML methodology

The ML approach highlighted in the proactive model not only eradicated NPD but also substantially slashed the overall number of tickets, resulting in a **cost-saving of 1 million USD.**

Prodapt

Approach to leverage ML model for ticket prediction and prioritization

ML Model Build Process Overview

Scaling the Data-Driven Frontier: A Step-by-Step Guide to Improve Predictive Network Fault Detection

Step: 1

Develop a pre and post-ticket logic to clarify the model's learning objectives

Step: 2

Incorporate a strategic threshold point in the alarm lifecycle to enhance data quality of the model

Step: 3

Integrate statistical modelling to improve the model's ticket learning while mitigating data biases

Step: 4

Utilize confusion matrix and metrics to infer ML model performance

The recommended implementation approach effectively overcomes challenges, creating an enhanced ML model that can predict which tickets need to be raised & prioritized. Additionally, it facilitates the grouping of tickets more quickly, leading to improved efficiency and effectiveness for the service providers.

Develop a **pre and post ticket logic** to clarify the model's learning objectives

Step: 1 A B 2 3 4

In Network Operations Centers (NOCs), **Pre-ticket logic is pivotal for forecasting alarm-to-ticket conversion, facilitating proactive issue management.** In contrast, **Post-ticket logic ensures sustained network health through maintenance and prevention measures.** Implementing these logical approaches is essential for ML model learning and operational efficiency while minimizing the risk of service disruption due to network issues.

Network element manager

- Router
- Gateway
- Switch

Network outage anomaly detection at NOC

Monitoring tools detect anomalies and trigger alarm

Alarm generation due to network anomalies

Alarm Trigger

Alarm Data	
Alarm Stage	One trigger equates to one alarm stage
Alarm Lifecycle	A combination of 'n' stages constitutes the alarm lifecycle
Alarm Type	Defines the purpose of the alarm

Network devices promptly **broadcast alarms during outages or anomalies**, ensuring NOC operators instantly receive and store them locally. Automated ticket generation **with ITOM tools accelerates issue resolution** and maintains detailed records, streamlining incident management.

1. Publish alarm trigger data

The network device triggers an alarm, signaling an outage or anomaly, and logs it in the cloud

2. Receive and store alarm trigger data

NOC is configured (Subscriber Code) to receive alarm-triggered data from the data cluster and store it in the local database at regular intervals

3. Ticket Generation

Utilizing ITOM tools, the system automatically generates tickets based on business rules

Group raw data into categories of alarm lifecycle occurrences and remove duplicates

Capture alarm dataset from ITOM portal

Capture the raw ticket data of alarms triggered by network devices due to outages or anomalies

Grouping Alarm Lifecycle

- Ticket Number
- Ticket State
- Action State
- Service Affected

Pre and Post Ticket Logic

Alarm stage tagged under ticket

Determine the earliest time at which the ticket was raised

Check if the alarm trigger time is earlier than the ticket's time

Yes

Pre-Ticket: Alarm stages have occurred before the ticket gets raised

No

Post-Ticket: Alarm stages have occurred after the ticket has been raised (reactive), and these will be removed from the dataset

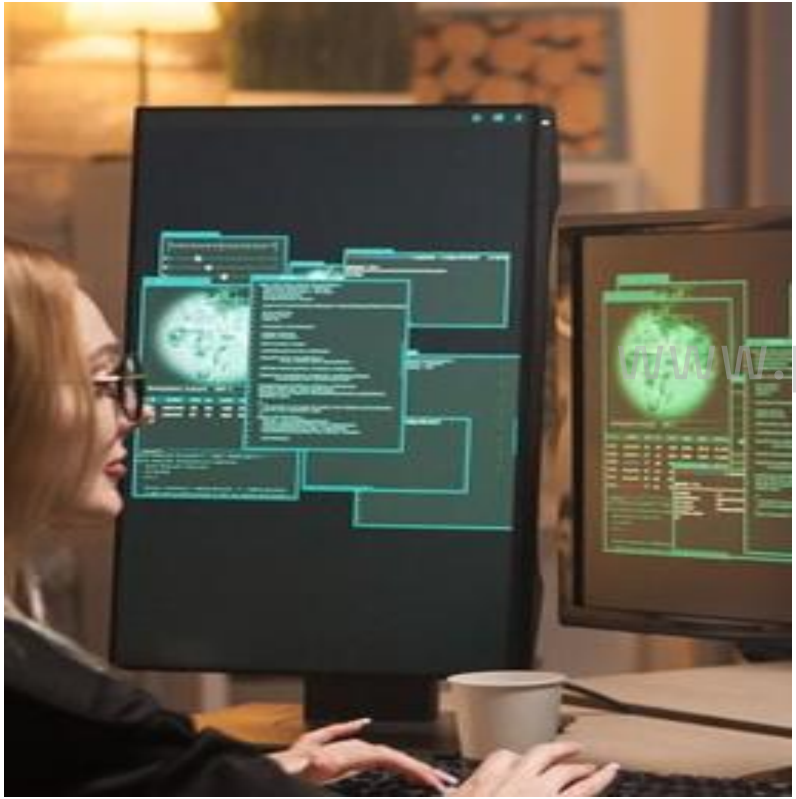
Processed data is fed to next step of data refining

Eliminated Dataset

Pre-ticket dataset & alarm trigger without ticket

Develop a **pre and post ticket logic** to clarify the model's learning objectives

Step: 1 A B 2 3 4



Recommendations

- Leverage BigQuery, a serverless data warehouse, to swiftly process massive data influxes, identifying patterns and potential threats in real-time, empowering the NOC to respond effectively to network challenges
- Utilize SQL for data preprocessing to boost memory efficiency, speed up processing, and cut down costs in data preparation for model training
- Mitigate outliers and missing data in the raw dataset through predictive analytics, enhancing data quality for precise ML model training and optimizing service engineer allocation to bolster network stability during disruptions such as high-attendance sports events
- Apply web scraping and APIs to extract external data, including alarm triggers and weather patterns, to reveal the correlation between network outages and adverse weather conditions, delivering valuable insights to service providers

Benefits

- Improved data quality with a 75% reduction in noisy and inadequate data for highly accurate model development
- Optimized model-building process &reduced costs through Google Cloud Platform's BigQuery for precision and efficiency

Incorporate a **strategic threshold point** in the alarm lifecycle to enhance the data quality of the model

Step: 1 2 A B 3 4

A strategic threshold point is pivotal for NOC to **minimize issues and unwarranted ticket generation**. To illustrate a service provider tackling alarms tied to signal strength fluctuations in cell towers. They identify a crucial point in the signal strength range through historical data analysis. Before reaching this point, the NOC takes prompt action to resolve issues. **Notably, the system prevents unnecessary alarms beyond this threshold, where critical issues diminish.** This proactive strategy optimizes resources, prevents network congestion, and ensures swift issue management, underscoring the indispensable role of strategic threshold in sustaining seamless service delivery.

Data Refinement Workflow: Improving model learning for improved network service delivery

Pre-ticket dataset & alarm trigger without ticket

Data preparation for cumulative analysis

Pre-ticket data is cleaned and transformed to derive insights, aiming to identify the strategic threshold point

Cumulative Occurrence Analysis

Identify that a significant majority of tickets are raised during the later stages of the alarm lifecycle

Apply the threshold point and exclude the alarms that have no ticket

Achieve Threshold Point

Alarm stage that doesn't have a ticket

Remove alarm stages beyond a critical point where a ticket is highly likely to occur

Data Noise Reduction in Alarm Triggers

Eliminate spammy false alarms that have been deprioritized

Deleting Spam Alarms

Extracting precise data which can lead to a ticket

Processed data is fed to the next step

Statistical Modelling and Sampling Techniques

"The workflow sharpens the dataset by precision-refining alarm data, strategically excluding noise, identifying key thresholds, and boosting ticket prediction accuracy"

Incorporate a **strategic threshold point** in the alarm lifecycle to enhance the data quality of the model

Step: 1 2 A B 3 4

Recommendations

- Employ conditional probability techniques to assess the probability of ticket generation at various alarm lifecycle stages, improving decision-making and resource allocation
- Implement statistical modeling methodologies to uncover more profound insights into the correlation between alarm lifecycle stages, ticket occurrence, and network downtime, improving decision-making and network engineer allocation
- Analyzing ticket patterns aids informed decision-making to prevent network congestion and downtime. In case of alarms for power failure, the system's shift to inverter or UPS support, along with a specified resolution timeframe, underscores the significance of a critical threshold value. These insights empower efficient issue management and network service reliability.

Benefits

- Achieved a 60% reduction in redundant data by removing irrelevant information, resulting in improved data quality, precision, modeling efficiency, and resource allocation
- Generates easily interpretable models using a noise-free dataset for improved understanding

Integrate **statistical modeling** to enhance the model's learning while mitigating data biases

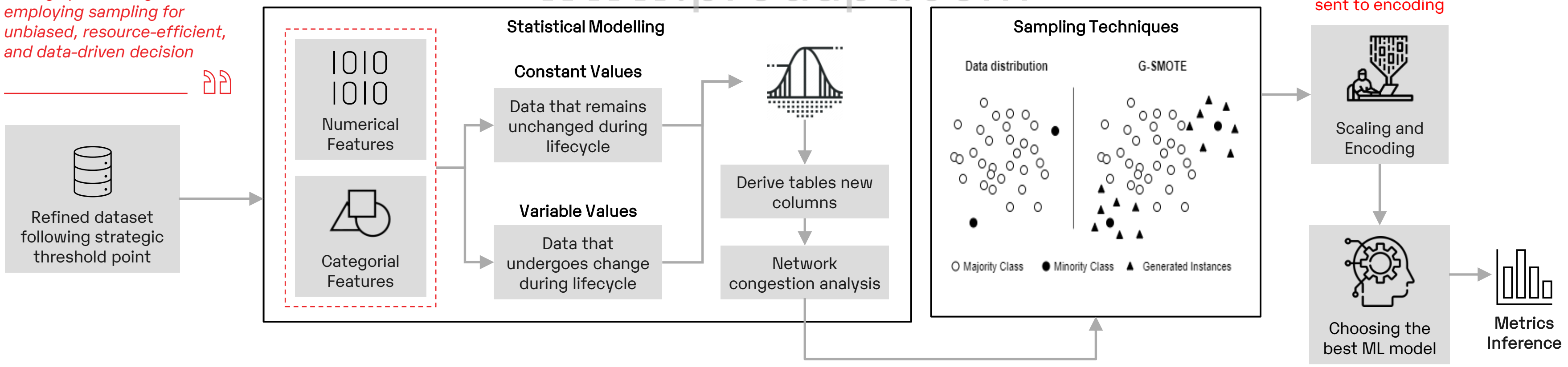
Step: 1 2 3 A B 4

In the network operations domain, statistical modeling **enhances network performance, reliability, and security by predicting patterns, optimizing resources, and addressing data imbalances.** Neglecting these steps may result in risks such as suboptimal outcomes, impaired network efficiency, and decision-making due to data biases.

Data-Backed Decisions Workflow: Mitigating data biases for improved network decision-making

Using statistical modeling, the NOC analyzes alarm and ticket data for insights like average processing time, employing sampling for unbiased, resource-efficient, and data-driven decision

Refining the dataset with feature categorization and the Synthetic Minority Over-sampling Technique (SMOTE) ensures fair and accurate ML predictions. This addresses imbalanced data, enhancing model robustness and reliability.



Integrate **statistical modeling** to enhance the model’s learning while mitigating data biases

Step: 1 2 3 A B 4

Recommendations

- Utilize oversampling techniques like SMOTE to create a balanced dataset by generating synthetic samples to address data biases. For example, if the network outage ticket type dominates the dataset, adding network performance as a sample ticket type can diversify the representation and mitigate data biases
- Utilize Vertex AI's standard scaling method for consistent feature scaling, fostering clean and balanced datasets, unbiased predictions, and improved network issue identification
- Improve the feature set by using time and stage data to generate new relevant features, enhancing the model's predictive capabilities, and capturing additional insights
- Create a classification ML model to predict ticket occurrence, facilitating informed decision-making and improved data organization

Benefits

- Achieved 90% sample validation to eliminate data bias, resulting in fairer and more impartial insights and predictions
- Attained dataset cleanliness by filtering out irrelevant data, thereby enhancing data quality
- Achieved a focused data distribution within a tight range, reducing variability and ensuring dataset balance

Utilize **confusion matrix and metrics** to infer ML model performance

Step: 1 2 3 4 A B

The **confusion matrix aids in assessing the performance of the ML model in predicting network issues**. For every ticket the ITSM system generates, the matrix can indicate whether the model has accurately predicted it (True Positive or True Negative) or made an error (False Positive or False Negative).

Confusion Matrix Model

Actual Tickets	Predicted Tickets: ML model predicts **NO TICKET** is required	Predicted Tickets: ML model predicts **TICKET** is required
No TICKET raised in the system	**TN** 2458K	**FP** 11K
TICKET raised in the system	**FN** 15K	**TP** 93K

The 4 Quadrants

- True Positives (TP): The model accurately predicted when a ticket was needed
- True Negatives (TN): The model accurately predicted when a ticket was unnecessary
- False Positives (FP): The model predicts a ticket needs to be raised, but no ticket was raised
- False Negatives (FN): The model predicts no ticket is required, but a ticket was raised

Interpreting the model's output

- High counts in TP and TN mean the model is working well
- High counts in FP and FN indicate areas for improvement
- Fine-tune the model to reduce FP or false tickets, causing unnecessary work
- FN counts need to be reduced to avoid truck roll-out

Actionable Insights for the Network team

- Take prompt action on FN
- TP: Investigate and resolve the network issue promptly to maintain service quality
- FN: Investigate the reason why the system is issuing an unnecessary ticket
- TN tickets require no action, and indicates a genuine issue or problem in the network that requires rectification

Utilize **confusion matrix and metrics** to infer ML model performance

Step: 1 2 3 4 A B

Recommendations

- Monitor key metrics like the recall rate, precision rate, and miss rate of the ML model to improve its prediction accuracy
- Utilize Python programming to efficiently code and develop ML models, as well as conduct data analysis and manipulation
- Employ process automation solutions to expedite routine tasks involved in ML model building, allowing service providers to allocate more time to model refinement and optimization
- Integrate APIs to seamlessly connect various external data sources and tools, enhancing data accessibility and accuracy for more effective model training and validation

Benefits

- The ML model identified numerous instances of unnecessary ticket generation, preventing truck dispatch and resulting in substantial operational cost savings of $1 million

Benefits achieved by a leading service provider in North America after implementation

90%
Improved Sample Validation

30%
Reduction in Network Downtime

10%
Ticket Reduction

$ 1 million
Reduction in OpEx

- No manual intervention required
- Enhanced application security and visibility
- Faster new feature deployment

Thank you

insights@prodapt.com

www.prodapt.com