06-04-2023

White Paper:
PMP Data Analysis

Grant Agreement No.:	101100680
Work Package:	WP6
Task Item:	Task 3
Nature of Document:	White Paper
Dissemination Level:	PU (Public)
Lead Partner:	CARNET
Document ID:	GN5-1-23-5BDD01
Authors:	Ljubomir Hrboka (CARNET)

Abstract

The Performance Measurement Platform (PMP) consists of a number of small nodes that undertake regular measurements towards a few Measurement Points (MPs) located in the core of the GÉANT network. This white paper presents an overview of the work and results of an analysis of the PMP perfSONAR monitoring data that was conducted to explore the possibilities of using machine learning techniques in the future.

Table of Contents

Table of Figures

Executive Summary

The Monitoring Task (Task 3) of the Network Development Work Package (WP6) in the GN5-1 project takes care of several network monitoring and management services in production. One of the services is the Performance Measurement Platform (PMP), which consists of a number of small nodes that undertake regular measurements towards a few Measurement Points (MPs) located in the core of the GÉANT network and operated by the GÉANT Network Operations Centre (NOC). An analysis of the PMP perfSONAR monitoring data was conducted to give quality insights into PMP measurement data and explore the possible usage of machine learning (ML) algorithms and other statistical methods to detect network anomalies. This paper presents an overview of the work and results of that analysis.

Exploratory data analysis (EDA) was performed on latency/jitter data, with possible outliers being identified from the observed datasets. Prospective directions for future work were identified.

1 Introduction

The Performance Measurement Platform (PMP) uses perfSONAR to undertake measurements. perfSONAR is an active network performance measurement tool that allows users to monitor, diagnose and troubleshoot their network performance. It uses a distributed network of measurement nodes to collect data on the performance of network services, such as file transfer and web access. This data can then be used to identify network bottlenecks and other performance issues, and to help network administrators optimise their network's performance.

Machine learning (ML) and artificial intelligence (AI) have many potential applications in networking, including network management, security, and optimisation. In network management, AI and ML algorithms can be used to monitor network traffic and identify potential issues or anomalies. This can help network administrators to proactively address problems and improve the reliability and performance of the network. AI can also be used to optimise network performance by automatically adjusting network resources in response to changes in demand or availability. Overall, the use of AI in networking has the potential to improve efficiency, reduce downtime, and enhance security. However, it is important to note that the implementation of AI in networking, like any other technology, can also introduce new risks and challenges that need to be carefully managed.

Part of the work within the PMP subtask of the Monitoring Task (Task 3) in the GN5-1 Network Development Work Package (WP6), ongoing from the GN4-3 project iteration, was to research the ways in which an analysis of historical perfSONAR data can be performed in order to detect network anomalies, pinpoint areas with ongoing issues, and support sensitive and/or high data traffic, particularly by the means of the modern machine learning techniques.

This paper describes the Performance Measurement Platform (Section 2), then presents the work and results of the PMP data analysis (Section 3), covering the exploratory data analysis (EDA) process, data availability, network latency outliers, measurement errors, and correlations between different types of measurements. It outlines two unsupervised techniques for anomaly detection and the concept of machine learning models (Section 4), and summarises key findings and potential future work (Section 5).

2 The Performance Measurement Platform

The Performance Measurement Platform (PMP) is set up as an open, trusted monitoring and measurement information infrastructure, provided to network engineers, Network Operations Centre (NOC) operators, research communities, network researchers and National Research and Education Network (NREN) participants to monitor, explore, practise and learn how network performance monitoring can contribute to better and more efficient usage and understanding of the existing multi-domain network infrastructure. It includes around 50 distributed measurement points with preinstalled perfSONAR which undertake regular measurements towards a few perfSONAR Measurement Points (MPs) located in the core of the GÉANT network (Figure 2.1). Performance data is stored in the central component, which therefore provides a large database of recorded measurements.

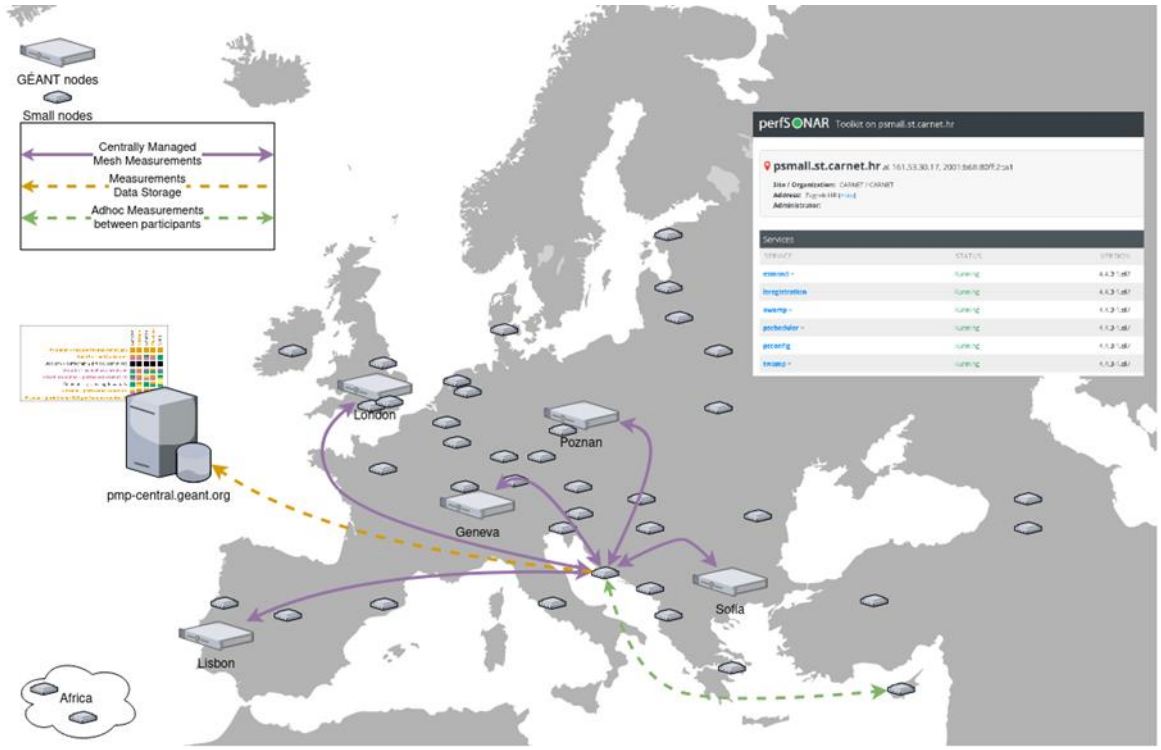

Figure 2.1: Performance Measurement Platform

Performance measurement data is stored at the central server located at pmp-central.geant.org. Different types of measurement results are available, gathered with different frequency. Most frequently executed are latency/jitter measurements, which are produced by the perfSONAR powstream application for continuous low-intrusion background one-way latency tests. Those results are available as histograms of about 600 latency values and one jitter value that are stored every minute for every link between small node and measurement point. In contrast, highly intrusive throughput measurements are conducted only a couple of times during the day. Other types of data available are Round-Trip Time (RTT) measurements configured at five-minute intervals, traceroute measurements configured at ten-minute intervals, and Hypertext Transfer Protocol (HTTP) response time and Domain Name System (DNS) transaction time measurements configured at one-hour intervals.

3 PMP Data Analysis

3.1 Exploratory Data Analysis

Exploratory data analysis (EDA) is a process of analysing and summarising a dataset in order to gain insights into and better understand the data. It is an important step in the data analysis process because it allows identification of patterns, trends, and relationships in the data that may not be immediately apparent.

EDA involves a number of different techniques and approaches, including:

1. Visualisation. Plotting data in various forms, such as histograms, scatter plots, and box plots, can help analysts identify patterns and relationships in the data.
2. Descriptive statistics. Calculating summary statistics, such as mean, median, and standard deviation, can help analysts understand the overall characteristics of the data.
3. Data cleaning. Removing or correcting errors or inconsistencies in the data can help ensure that the analysis is based on accurate and reliable data.
4. Data transformation. Transforming the data, such as by normalising or scaling it, can help analysts better understand the relationships between different variables.

The goal of EDA is to identify interesting and meaningful patterns in the data that can be further explored and analysed. It is an iterative process that allows analysts to ask and answer questions about the data, and to refine their understanding of it as they learn more.

3.2 Data Availability

The recommended way of reading and writing data from the perfSONAR measurement archive is the Esmond Application Programming Interface (API), a Representational State Transfer (REST) interface that allows researchers to query and access the data. REST APIs are a type of web service that allows applications to interact with data or functionality provided by the API over the web using a set of standard HTTP methods. The process to get the data using the Esmond API usually follows these two steps:

1. Find the measurements with the type of data that is being looked for.
2. Retrieve the results data for the measurements found.

In order to collect the data a number of simple Python scripts were developed. Scripts were fed with the small node endpoints, time period, and the type of measurement. Limited server memory and CPU resources prevented the central server from returning a massive amount of data, so additional techniques were developed in order to collect larger amounts of data for analysis. These techniques typically included querying the central server for limited time data, usually for a one-day period, collecting that data, repeating the query for the following day and merging the results into a single appropriate data object. That way the server was not overloaded and the goal to collect the data for the longer period was achieved.

One such example of the analysed latency samples is shown in Figure 3.1. The idea was to aggregate data between two endpoints for a two-month period in order to visually inspect the homogeneity of the data.

Normally, without any outside disturbance (e.g. link errors, hardware malfunctions, etc.), latency between two endpoints should remain constant. The data-collecting script performed sixty queries in order to get the data for the specified period, resulting in a single file containing thirteen million latency records. During the process of data exploration, it was visually observed that outliers tend to group around several distinctive values. Since the number of outliers was relatively small compared with the whole dataset, logarithmic scale was used to emphasise those points.

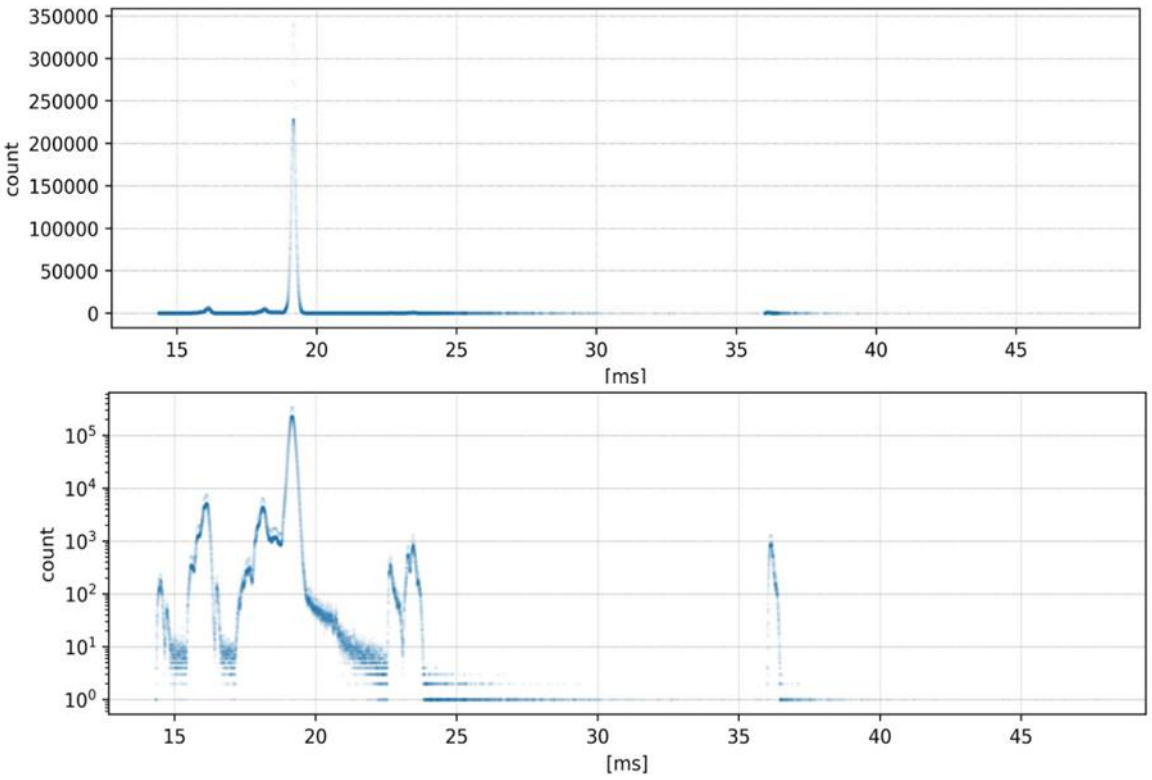

Figure 3.1: Latency distribution of around 13 million samples between two endpoints in a two-month period presented in linear and logarithmic scale

3.3 Network Latency Outliers

Network latency is important to consider in many applications, such as video conferencing and other real-time communication systems, as it can affect the quality and responsiveness of the service. Network latency between two nodes would ideally be constant over time but due to network congestion, hardware issues, changes in the number of hops and various other factors it can fluctuate. While small fluctuations in the network latency are normal and various applications are usually developed to deal with them, bigger fluctuations, such as doubling the delay, can have a huge impact on the performance.

Figure 3.2 represents what would be called “normal” latency distribution. As can be observed from the time sequence of packets and from the corresponding histogram, in the sample of a thousand packets maximal deviation from the average latency is within several percentage points.

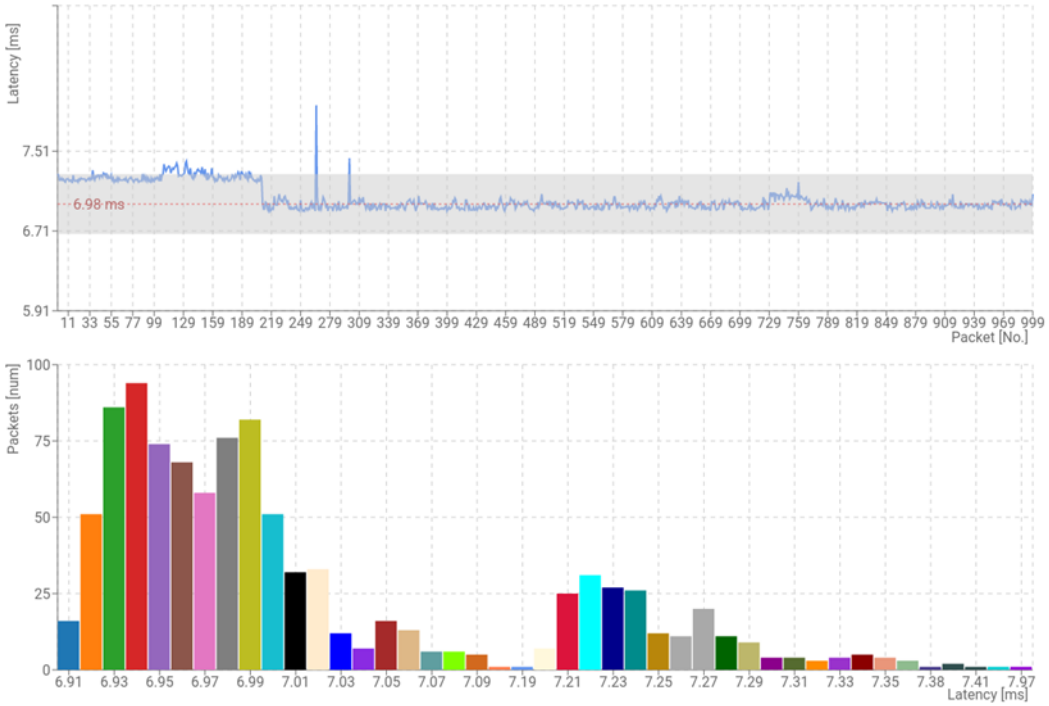

Figure 3.2: Normal fluctuation in latency

On the other hand, Figure 3.3 presents an anomalous event that happened on another stream of a thousand consecutive packets. On the time sequence graph it can be observed that, for an interval of about three hundred packets, average latency was almost three times higher than average latency on that link and that the histogram is heavily skewed to the right. Such an event could result in a noticeable degradation in some services.

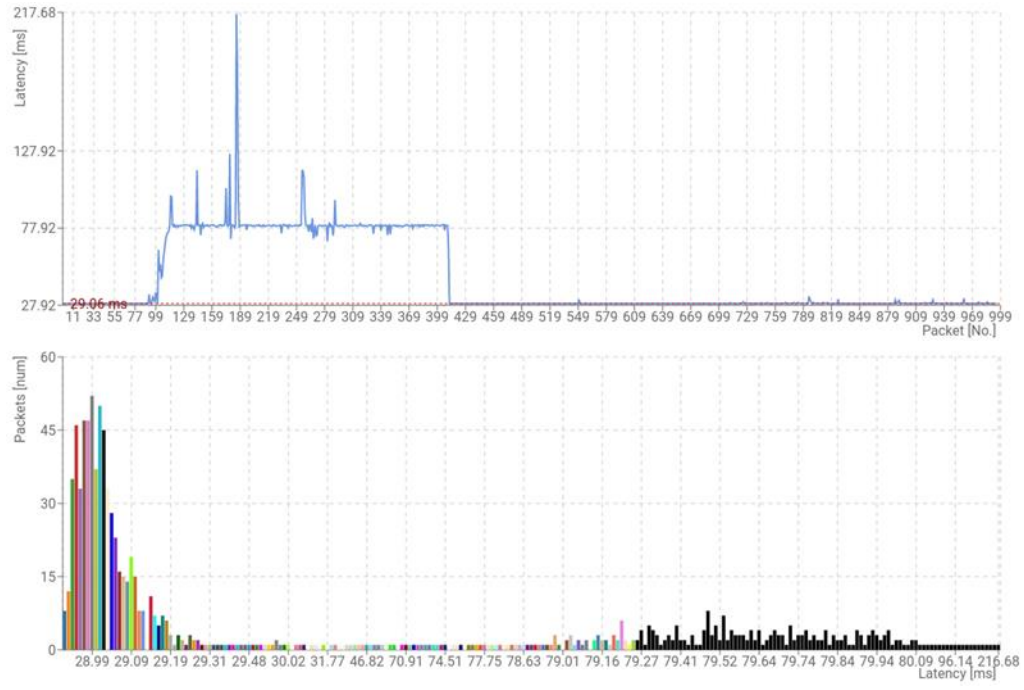

Figure 3.3: Anomalous fluctuation in latency

Visual analysis of network latency distributions between different endpoints has shown that aforementioned anomalous situations where the latency doubles or triples during some time periods occur more than normally expected. A couple of examples are given in Figure 3.4 and Figure 3.5, where network latency fluctuates. In Figure 3.4 such fluctuations are noticeable around 45 ms, and in Figure 3.5 around 36 ms.

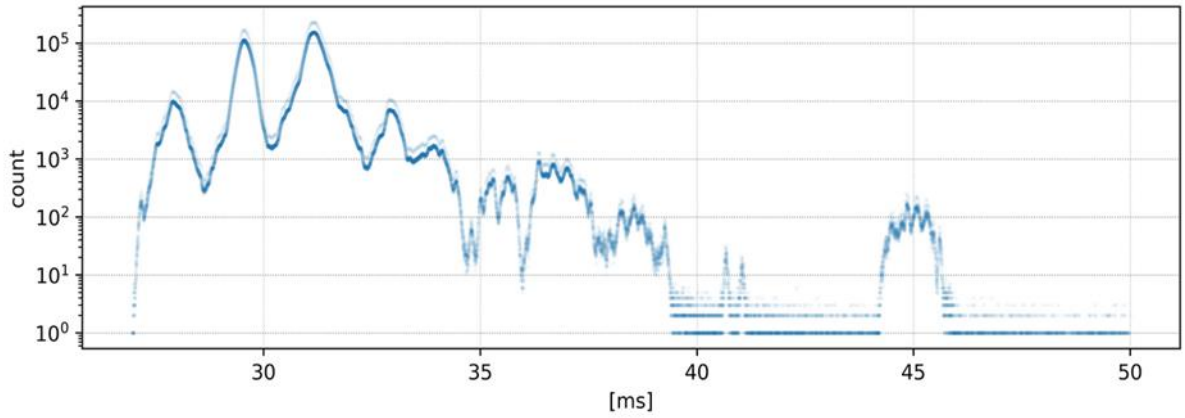

Figure 3.4: Multiple outliers in distribution, noticeable grouping around 45 ms value

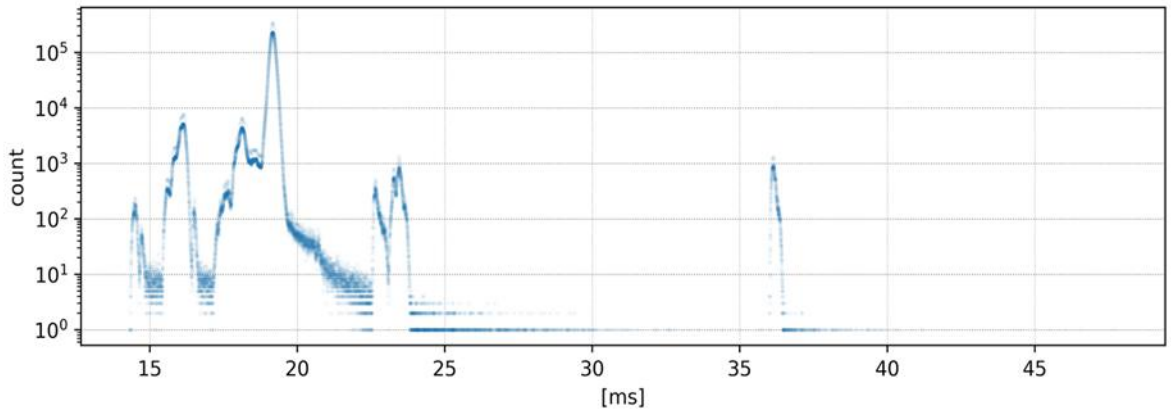

Figure 3.5: Multiple outliers in distribution, noticeable grouping around 36 ms value

3.4 Measurement Errors

On several generated network latency distribution graphs, measurement errors were clearly visible. For example, in some cases measured latency was only a few milliseconds, as can be seen in Figure 3.6.

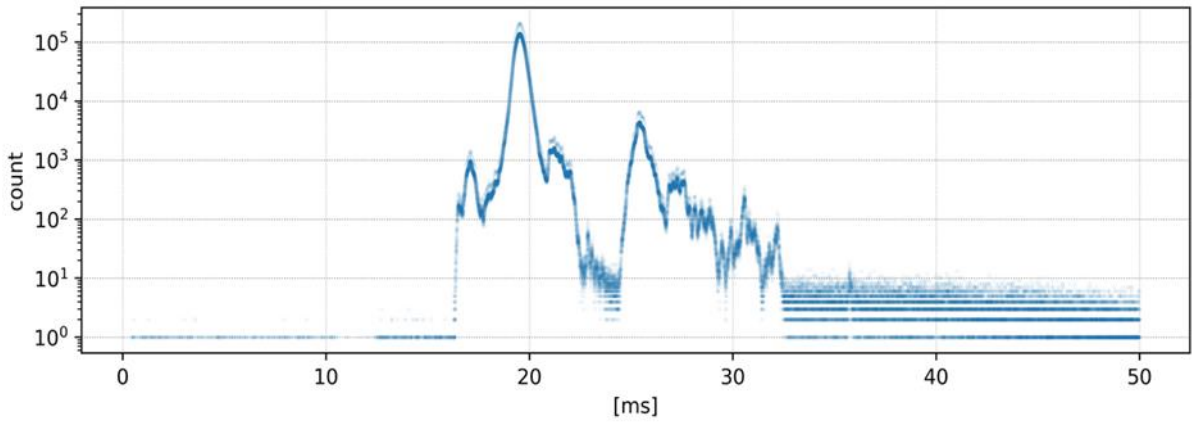

Figure 3.6: Questionably small values of network latency in the range 0–10 ms

This is an obvious error, because it is physically impossible for network latency to be only a couple of milliseconds on a link where average values are between 20 and 30 milliseconds. In some cases measured latency is shorter than the optical signal propagation time between the two nodes. It can be speculated that the root cause of this anomaly is clock drift on measurement nodes or some kind of bug in the perfSONAR software.

3.5 Correlations between Different Types of Measurements

Although delay/jitter were the measurements used, the idea that these measurements can be correlated with other types of measurement and meaningful results produced was also tested. A couple of events were visually examined in detail to check whether correlation can be detected. In this example, points of interest were the first two points from the throughput graph where a major throughput anomaly was observed. The basic idea was to determine whether an anomaly in latency/jitter can be seen at the exact moment at which a visible throughput anomaly existed. As mentioned above, latency/jitter results were saved every minute, while throughput measurements were performed 3–4 times a day.

Jitter measurements in the interval of ten minutes before and ten minutes after the observed throughput anomaly were pulled out from the set of measurement results and displayed with T_0 (exact moment of the throughput anomaly) placed in the centre. Jitter measurements from the first examined anomaly in Figure 3.7 show no noticeable outliers since results are scattered and there is no value that stands out significantly. On the other hand, jitter measurement results obtained in a similar way for the moment T_0 of the second throughput anomaly (Figure 3.8) show a clear outlier exactly after the throughput test was performed. The first jitter measurement after the moment of the throughput anomaly is more than double the previous values and the next one is ten times higher. After that moment, jitter measurement results return to their normal values, indicating that two events (jitter and throughput measurement outliers) correlate with each other.

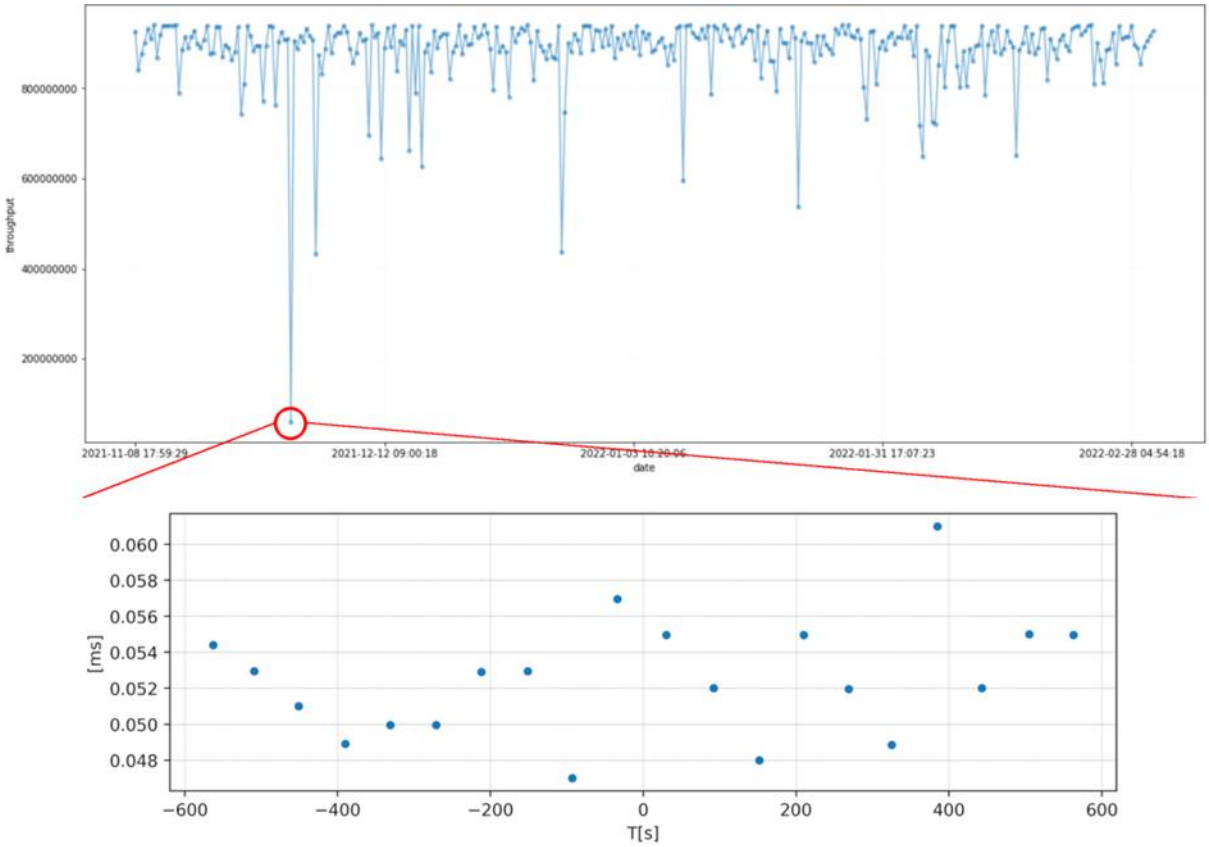

Figure 3.7: No visible jitter outlier at the moment of throughput anomaly

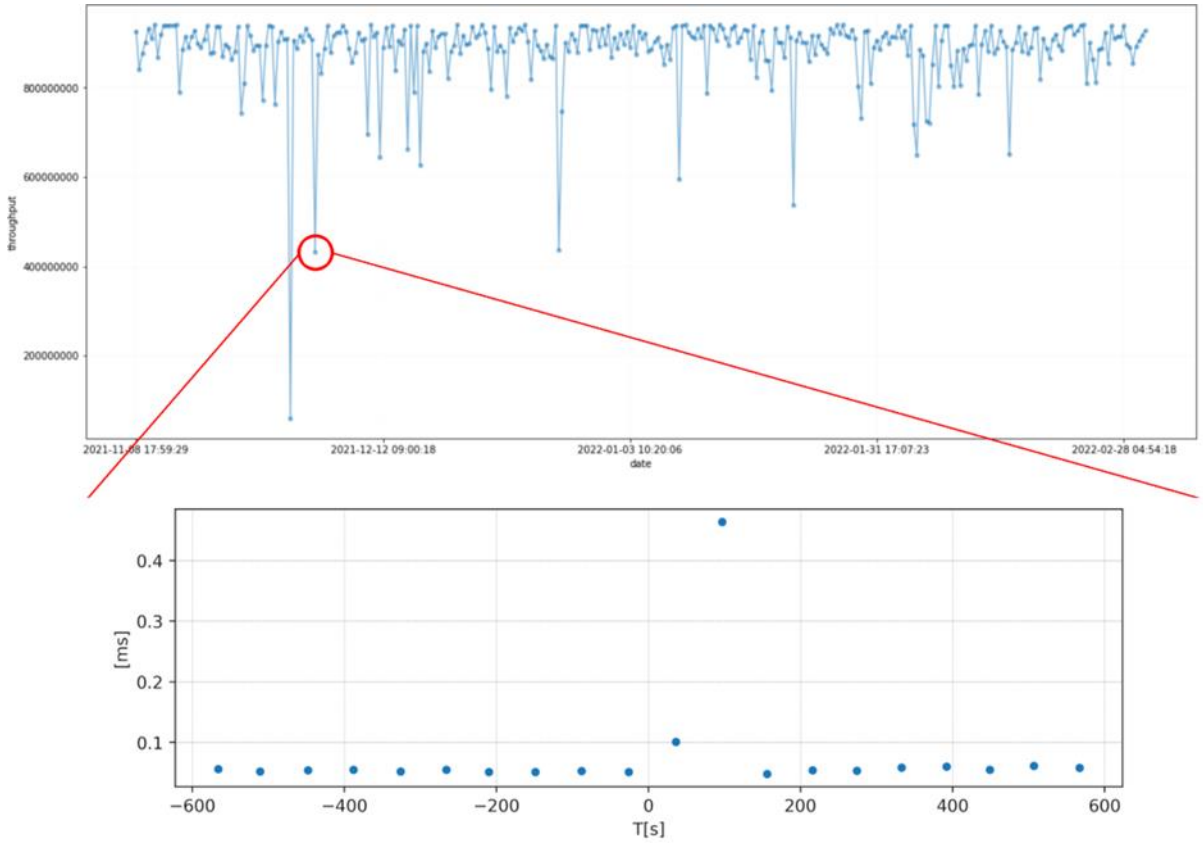

Figure 3.8: Visible jitter outlier at the moment of throughput anomaly

These examples show that there is no clear rule that an anomaly in one parameter may necessarily be directly related to anomalies in another parameter. Therefore, it is important to conduct further investigation to understand the underlying causes and relationships between different parameters for each anomaly detected. Furthermore, misalignment of the frequency of measurements can create challenges in cross-parameter analysis, requiring a careful approach to ensure accurate and meaningful data interpretation.

4 Machine Learning Models

Since there is no labelled training data, unsupervised learning algorithms are currently the best option to experiment with anomaly detections. Principal Component Analysis (PCA) and autoencoders [Brauckhoff], [Chen], [Kiran] are two different unsupervised techniques that can be used for anomaly detection.

PCA is a classic technique that works by identifying the directions in which the data varies the most and projecting the data onto these directions, effectively reducing the number of dimensions while retaining as much of the variation in the data as possible. The technique would normally be implemented to analyse available data and determine what would be a "normal" representation and then calculate the distance metric to each new data point in order to determine the anomaly.

Autoencoders are a type of neural network that are trained to reconstruct the original input data from a lower-dimensional representation. Autoencoders consist of an encoder network that maps the input data to a lower-dimensional representation and a decoder network that maps the lower-dimensional representation back to the original data. The encoder and decoder networks are trained jointly, with the goal of minimising the reconstruction error between the original data and the reconstructed data.

The aim of this analysis was to find "normal" data without any visible anomalies. Data ought to be cleaned, all of the anomalies and outliers removed, and only data from visually "normal" distributions are used as a training dataset.

Also, the idea was to do this for all end-to-end links and to develop this into an ML automated pipeline (Figure 4.1) with a model registry and data pipeline to perform continuous analysis on measurement data.

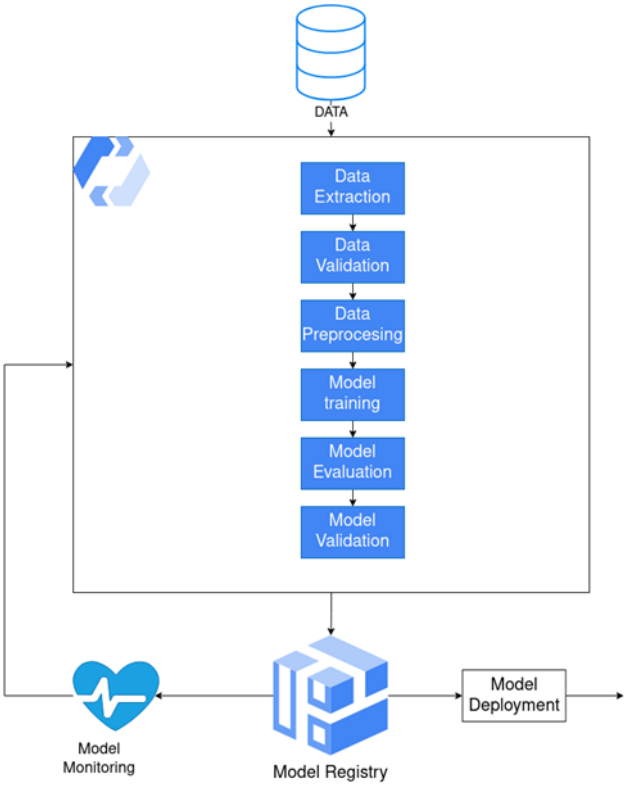

Figure 4.1: ML automated pipeline

A machine learning model registry is a system that stores and manages machine learning models. It allows tracking, versioning and deploying of machine learning models in a centralised and organised way. Such a system can manage the selection and the lifecycle of machine learning models more effectively and ensure that the most accurate and up-to-date model is being used for every end-to-end link.

During this phase of the project, autoencoders and PCA analysis were only experimented with, but might be considered as an area of potential future work. There are also some statistical methods such as chi-squared distance between histograms that can be tested for comparison and efficiency.

5 Conclusions

perfSONAR historical data kept on the PMP central server is a unique collection of performance measurement data that can be used and analysed in order to detect anomalies and deteriorating conditions within the GÉANT network. The PMP central server allows a holistic view of network performance and provides a perfect place to use machine learning techniques that can lead to faster and more accurate anomaly detection, improved decision-making, improved accuracy, and cost reduction.

Working on the development of scripts and other methods for retrieving data allowed us to understand the PMP system better, the way data is stored in it, and expectations of what could be obtained from it. Exploratory data analysis that was performed as part of this work gave quality insights into measurement data and confirmed that it can be used for the purpose of anomaly detection. Insights into measurement data are a very important part of choosing and building a machine learning model. This work provided proof of concept for a number of ways anomalies in measurement data can be detected, regardless of whether they originate from real events or measurement errors. The possibility of correlation between different measurements was also proven, but that type of analysis should be conducted with caution since it was shown that if measurements of different parameters are taken at different times or with different frequency, it may be challenging to compare and analyse the data effectively.

The future continuation of this work could be focused on the development of an operational system for automatic near-real-time detection of anomalies based on an ML model registry and PMP data for different end-to-end points.

References

[Brauckhoff] Brauckhoff, Daniela; Salamatian, Kave; May, Martin. “Applying PCA for Traffic Anomaly Detection: Problems and Solutions”. *IEEE INFOCOM 2009*, pp. 2866–2870. DOI: 10.1109/INFOCOM.2009.5062248.

[Chen] Chen, Zhaomin, et al. “Autoencoder-based network anomaly detection”. *2018 Wireless telecommunications symposium (WTS)*, pp. 1–5. DOI: 10.1109/WTS.2018.8363930.

[Kiran] Kiran, Mariam, et al. “Detecting anomalous packets in network transfers: investigations using PCA, autoencoder and isolation forest in TCP”. *Machine Learning* 109, 1127–1143 (2020). https://doi.org/10.1007/s10994-020-05870-y.

Glossary

AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
DNS	Domain Name System
EDA	Exploratory Data Analysis
HTTP	Hypertext Transfer Protocol
ML	Machine Learning
NOC	Network Operations Centre
NREN	National Research and Education Network
PCA	Principal Component Analysis
perfSONAR	performance-focused Service Oriented Network monitoring ARchitecture
PMP	Performance Measurement Platform
REST	Representational State Transfer
RTT	Round-Trip Time
WP	Work Package
WP6	Work Package 6 Network Development