Journal of Computer and Communications, 2023, 11, 122-141
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

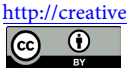
Machine Learning-Based Alarms Classification and Correlation in an SDH/WDM Optical Network to Improve Network Maintenance

Deussom Djomadji Eric Michel[1,2], Takembo Ntahkie Clovis[1], Tchapga Tchito Christian[1], Arabo Mamadou[2], Michael Ekonde Sone[1]

[1]Department of Electrical and Electronic Engineering, College of Technology, University of Buea, Buea, Cameroon
[2]Division of Information and Communications Technology, The National School of Posts and Telecommunications and Information and Communication Technologies, University of Yaoundé I, Yaoundé, Cameroon
Email: eric.deussom@gmail.com

How to cite this paper: Michel, D.D.E., Clovis, T.N., Christian, T.T., Mamadou, A. and Sone, M.E. (2023) Machine Learning-Based Alarms Classification and Correlation in an SDH/WDM Optical Network to Improve Network Maintenance. *Journal of Computer and Communications*, **11**, 122-141.
https://doi.org/10.4236/jcc.2023.112009

Received: January 26, 2023
Accepted: February 25, 2023
Published: February 28, 2023



Abstract

The evolution of telecommunications has allowed the development of broadband services based mainly on fiber optic backbone networks. The operation and maintenance of these optical networks is made possible by using supervision platforms that generate alarms that can be archived in the form of log files. But analyzing the alarms in the log files is a laborious and difficult task for the engineers who need a degree of expertise. Identifying failures and their root cause can be time consuming and impact the quality of service, network availability and service level agreements signed between the operator and its customers. Therefore, it is more than important to study the different possibilities of alarms classification and to use machine learning algorithms for alarms correlation in order to quickly determine the root causes of problems faster. We conducted a research case study on one of the operators in Cameroon who held an optical backbone based on SDH and WDM technologies with data collected from 2016-03-28 to "2022-09-01" with 7201 rows and 18. In this paper, we will classify alarms according to different criteria and use 02 unsupervised learning algorithms namely the K-Means algorithm and the DBSCAN to establish correlations between alarms in order to identify root causes of problems and reduce the time to troubleshoot. To achieve this objective, log files were exploited in order to obtain the root causes of the alarms, and then K-Means algorithm and the DBSCAN were used firstly to evaluate their performance and their capability to identify the root cause of alarms in optical network.

Keywords

Optical Network, Alarms, Log Files, Root Cause Analysis, Machine Learning

1. Introduction

The evolution of telecommunication networks, in particular optical networks based on Synchronous Digital Hierarchy (SDH) and Wavelength Division Multiplexing (**WDM**) technologies [1] [2], has marked a major transformation of the world by enabling the massive sharing of information at high speed across the globe. Over the past decades, this hyper-connectivity has become deeply embedded in our daily lives and in most areas of activity. As a result, telecom operators must be able to ensure the smooth operation of these networks. To achieve this, it is essential to be able to efficiently identify the failures that occur and their origin.

Root cause analysis consists in finding the primary cause of a failure amongst a multitude of errors. Due to the complexity of networks, both in size and in technology, finding the root cause of a failure remains a difficult task. Today, the emergence of artificial intelligence with applications such as machine learning allows for a better root cause analysis. This can be possible by analyzing alarms logs. But to be able to implement these methods and end up with a maintenance aid tool, it is necessary to acquire a huge amount of data on the operating status of the optical network equipment. The acquisition of data will allow to train the machine learning algorithms [3] [4] [5] and to increase their performances.

It is within this framework that this article aims to implement a machine learning model that will facilitate the analysis of data from the log files of the optical network equipment from the Operation and Maintenance Centre (OMC) supervision servers of an operator. It will be a matter of recording the data extracted from the server, then performing a classification and a correlation of the alarms and determining the root causes. In fact, network maintenance almost always relies on alarm monitoring and management to identify existing problems and provide solutions. The proposed solutions must be implemented as quickly as possible in order to reduce the time of degradation or interruption of services. For optical networks, operators in the field sign Service Level Agreements (SLAs) with their customers and these SLAs must be respected otherwise financial compensation must be made by the operator to its customers. It is therefore important to find new approaches to help the rapid identification of network failures or various defects. Therefore, alarm classification and machine learning tools [6] [7] [8] are very useful to optimize network availability and increase the overall quality of service offered to customers. Several authors have focused on alarm classification approaches or on the use of machine learning to solve problems in networks [9] [10] [11]. We have for example E. DEUSSOM et al in ref. [9] who worked on fraud detection in mobile networks by analysing Call Data Records (CDRs) and traffic; moreover, B. BATCHAKUI et al in ref. [10] worked on the comparison of machine learning algorithms to improve the maintenance of Long Term Evolution (LTE) Time Division Duplexing (TDD) networks. M. Klemettinen, H. Mannila, and H. Toivonen in ref. [11] proposed a method for discovering recurring patterns of alarms in databases using a corre-

lation system. They also present a tool with which network management experts can browse the large amounts of rules produced.

T. White and N. Ross in ref. [12] proposed architecture for an alarm correlation engine. They design a correlation engine directly into the Network Management System (NMS). The idea is to improve some aspects of the NMS to reduce the observed raw flow of alarms by sending only the most relevant information. Some methods define an alarm architecture using the Model-Based Reasoning principle. These methods introduce rule-based approaches to group alarms that refer to the same problem.

A. Bouillard, A. Junier, B. Ronot in ref. [13] proposed a method that creates several alarm dependency graphs, based on the sequence of alarm names, which allows a quick study of the alarm correlation problem through a powerful heuristic algorithm.

N. Amani, M. Fathi, M. Dehghan in ref. [14] proposed a new Case-Based Reasoning (CBR) method for alarm correlation in telecommunication networks. The proposed method has been simulated by developing three main modules: a module for generating faults and alarms, defining the network configuration, and filtering and correlating alarms using the CBR. One of the most important aspects of the results obtained was the speed of the system.

In this work, we conducted a research case study on one of the operators in Cameroon who hold an optical backbone based on SDH and WDM technology with data collected from 2016-03-28 to "2022-09-01" with 7201 rows and 18. The method used in this study uses machine learning with unsupervised learning techniques which are quite popular methods used in alarms classification and correlation. The purpose of this study is to propose an effective method that can be applied to detect root cause of problems based on alarm logs analysis. The rest of the paper is organized as follow: Materials and methods will be presented in the second section. The third section deals with the experimentation carried out and the results obtained; the fourth section presents the discussion of the results obtained. Finally, a general conclusion and perspectives are presented.

2. Materials and Methods Used

In the context of the present work, the research was done by using real data collected on an optical network management platform in Cameroon using Huawei Network Cloud Engine management platform (NCE-Transmission). The Topology management and alarm management feature of the NCE-Tx was selected to monitor the network elements, network alarms and process the alarms based on their severity, types, sources, and impact on the network (see **Figure 1** and **Figure 2** which present the appearance the network entities through the topology management and the alarms on the alarm management platform of the NCE-Tx). Hence, the target data to be predicted are discrete data. Algorithms used for the classification of the collected data, as well as for the prediction of our target are presented and reviewed below: K-Means and DBSCAN.

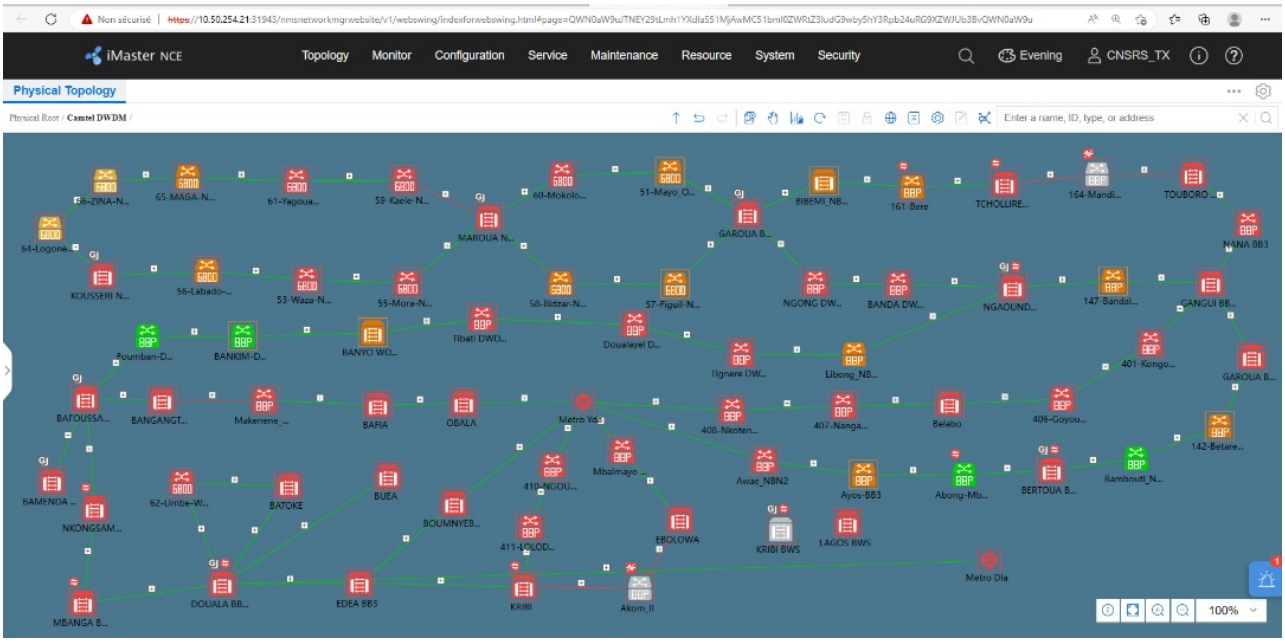


Figure 1. Structure of NCE-Tx topology management.

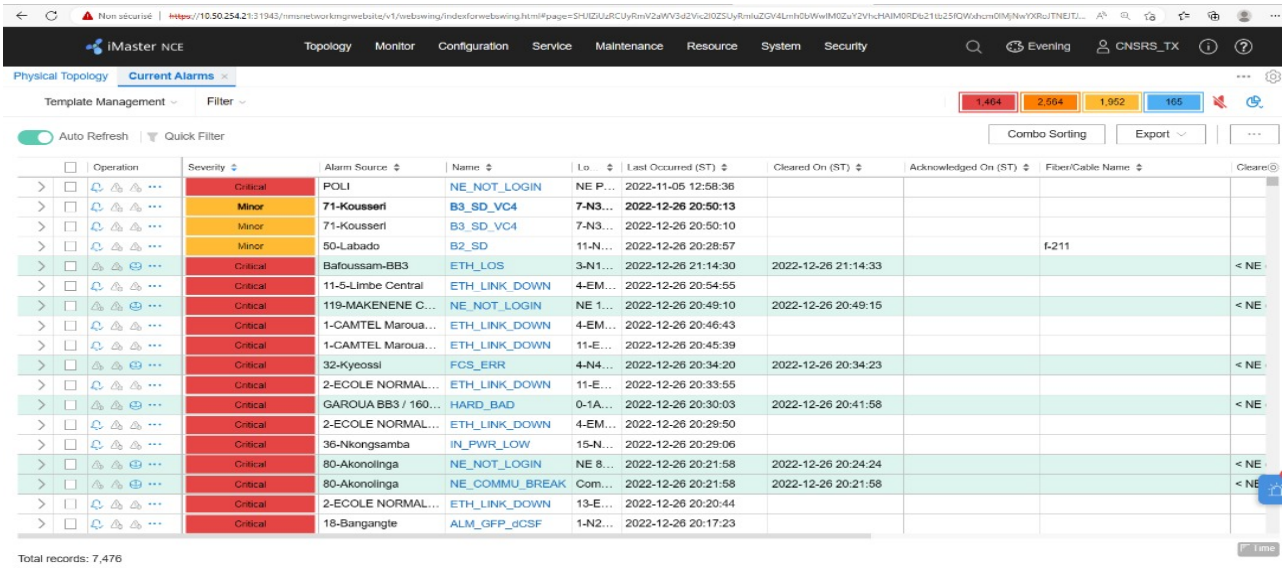


Figure 2. Structure of NCE-Tx Alarm management system.

In the context of this work, the target data to be exploited are unlabeled data. Thus, we will present the algorithms used for the classification of our data. These are the K-Means and DBSCAN algorithms.

2.1. K-Means Algorithm

K-means is an unsupervised non-hierarchical clustering algorithm. It allows to group in K distinct clusters the observations of the dataset. Thus, similar data will be found in the same cluster. Moreover, an observation can only be found in one cluster at a time (exclusivity of membership). The same observation cannot

belong to two different clusters [15]. In order to group a dataset into K distinct clusters, the K-Means algorithm needs a way to compare the degree of similarity between the different observations. Thus, two data that are similar will have a reduced dissimilarity distance, while two different objects will have a larger separation distance. The K-means method is quite simple, starting with the selection of the number of clusters as many as K pieces, and then K pieces of data are randomly taken from the data set as a centroid to represent a cluster. All data are then calculated at the distance of the centroid and each data will be a member of a cluster represented by a centroid that has the closest distance to the data. Finally, the re-calculation of the centroid value obtained from the average value of each cluster [16].

2.2. DBSCAN Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a well-known unsupervised algorithm in clustering. DBSCAN is a simple algorithm that defines clusters using local density estimation [17].

In order to train the system to be able to determine the different clusters and to propose the most optimal unsupervised classification, we followed an approach borrowed from "data science". Indeed, from a set of data (dataset) collected from the databases of the operation and maintenance center of a cell phone operator during a period, we have executed on the said data, the machine learning algorithms presented in the previous paragraphs, in order to select the algorithm that will produce the most expected results. The approach consists of:

- Processing the dataset;
- Normalization/coding of the variables;
- Determination of the model and its parameters;
- Learning;
- Test and interpretation of the results.

2.3. Environment and Tools Used

To run the machine learning algorithms, we used the Python programming language, in particular version 3.8. It is the reference language used in the development of applications for artificial intelligence. It is easy to install, uncompiled, fast and light. The Python distribution used is Anaconda: it contains all the tools and libraries we need to do machine learning: Numpy, Matplotlib, Sklearn, Jupiter, Spider…etc.

2.3.1. Dataset Processing

The dataset is the set of examples that the machine must study. Our data were collected directly from the operator's optical network monitoring platform system. Here we will focus on the alarm log file, whose entries are grouped into a single file. This is a plain text file in which each entry is recorded on one and only one line. The content of each line is however organized according to a nomenclature that can be configured by the user. Studying this nomenclature be-

fore the automatic analysis of the logs makes it possible to gain in efficiency (because the information is easily identifiable) and in time (because the analysis is then done on smaller chains). This information is among others the severity of the alarm, its identifier, the name of the failures, the location of the failure, the number of occurrences, etc… **Figure 3** is an extract of the data constituting our dataset.

We have a database with 7201 rows and 18 columns, 15 of which are of Type Object and 3 of Type Integer (Int64). The following python code gives the list of columns of the dataset as presented in **Figure 4**.

Before proceeding to the coding of the clustering algorithm of the solution to be applied with the help of the log files, we carried out some analyses with the tools offered by the ANACONDA environment, through its libraries Numpy and Matplotlib. These analyses allowed us to determine on the one hand, the columns or features to use, and the degree of correlation between the ALARM_ID parameter and the other parameters of the database. **Figure 5** presents the code used for the transformation of the values of the column "First occurred" in ms.

Maintenance Status is in NORMAL.																		
	Severity	Alarm ID	Name	Alarm Sou	Location I	Occurrenc	First Occur	Last Occur	Cleared O	Acknowlec	Fiber/Cabl	Cleared By	Acknowlec	Clearance	Acknowlec	Alarm Seri	Maintenar	Affected Service
-	Minor	63592	LP_REI_VC	Nana-SDH	15-N5EFSC	1034	2022-08-3	2022-09-0	-	-	-	-	-	Uncleared	Unacknow	20893289	NORMAL	-
-	Critical	235	ETH_LOS	Bafoussan	3-N1EFS0A	101	2022-08-3	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20893232	NORMAL	-
-	Major	63638	ALM_GFP_	Nana-SDH	15-N5EFSC	16	2022-08-3	2022-09-0	-	-	-	-	-	Uncleared	Unacknow	20893248	NORMAL	-
-	Minor	63592	LP_REI_VC	Nana-SDH	15-N5EFSC	1162	2022-08-3	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20893209	NORMAL	-
-	Minor	63592	LP_REI_VC	Nana-SDH	15-N5EFSC	1189	2022-08-3	2022-09-0	-	-	-	-	-	Uncleared	Unacknow	20893198	NORMAL	-
-	Minor	63592	LP_REI_VC	Nana-SDH	15-N5EFSC	1169	2022-08-3	2022-09-0	-	-	-	-	-	Uncleared	Unacknow	20893187	NORMAL	-
-	Minor	63592	LP_REI_VC	Nana-SDH	15-N5EFSC	1079	2022-08-3	2022-09-0	-	-	-	-	-	Uncleared	Unacknow	20893148	NORMAL	-
-	Critical	8	R_LOF	MOKOLO_	6-N1SL1-1	2	2022-08-3	2022-09-0	-	-	-	-	-	Uncleared	Unacknow	20893290	NORMAL	-
-	Minor	63591	LP_RDI_VC	10-1-Bepa	13-N3EGS	1	2022-09-0	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20892862	NORMAL	-
-	Minor	63591	LP_RDI_VC	10-1-Bepa	13-N3EGS	1	2022-09-0	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20892862	NORMAL	-
-	Minor	63591	LP_RDI_VC	10-1-Bepa	13-N3EGS	1	2022-09-0	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20892862	NORMAL	-
-	Minor	63591	LP_RDI_VC	10-1-Bepa	13-N3EGS	1	2022-09-0	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20892862	NORMAL	-
-	Minor	63591	LP_RDI_VC	10-1-Bepa	13-N3EGS	1	2022-09-0	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20892861	NORMAL	-
-	Minor	63591	LP_RDI_VC	10-1-Bepa	13-N3EGS	1	2022-09-0	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20892861	NORMAL	-
-	Minor	63591	LP_RDI_VC	10-1-Bepa	13-N3EGS	1	2022-09-0	2022-09-0	2022-09-0	-	-	< NE opera	-	Cleared	Unacknow	20892861	NORMAL	-

Figure 3. Data extract from the dataset.

```
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0                             7195 non-null   object
 1   Severity                  7195 non-null   object
 2   Alarm ID                  7195 non-null   int64
 3   Name                      7195 non-null   object
 4   Alarm Source              7195 non-null   object
 5   Location Info             7195 non-null   object
 6   Occurrences               7195 non-null   int64
 7   First Occurred (ST)       7195 non-null   object
 8   Last Occurred (ST)        7195 non-null   object
 9   Cleared On (ST)           7195 non-null   object
 10  Acknowledged On (ST)      7195 non-null   object
 11  Fiber/Cable Name          7195 non-null   object
 12  Cleared By                7195 non-null   object
 13  Acknowledged By           7195 non-null   object
 14  Clearance Status          7195 non-null   object
 15  Acknowledgement Status    7195 non-null   object
 16  Alarm Serial Number       7195 non-null   int64
 17  Maintenance Status        7195 non-null   object
 18  Affected Service          7195 non-null   object
dtypes: int64(3), object(16)
memory usage: 1.0+ MB
```

Figure 4. Dataset columns.

```
df["First Occured in ms"]= df["First Occurred (ST)"].apply(lambda x:
datetime.strptime(x, '%Y-%m-%d %H:%M:%S').timestamp() * 1000 )
```

Figure 5. Transformation of the values of the column First occured in ms.

We transformed the data in order to make them more usable. These are:

- Transformation of the column "first occurred (ST)" into ms.
- Assign the IDs to the elements of the "Location Info" column, from 0 to 4761;
- Assign the IDs to the elements of the "Alarm Source" column, from 0 to 282;
- Assign the IDs to the elements of the "Severity" column from 0 to 3.

The analysis of correlations through the heatmap (Figure 3) allows to visually representing correlations (or relationships) between variables.

sns.heatmap(corr, cmap = "RdBu", vmin = −1, vmax = 1, annot = True)

From Figure 6 below, we can observe weak correlations between Alarm ID and the other parameters:

- **Number of occurrences**;
- **Date of the first occurrence**;
- **Location ID**;
- **Source ID**.

We can see that the level of severity is not linked to other parameters. We will now classify these different alarms.

2.3.2. Classification of Alarms According to Various Criteria

Figure 7 below presents the data header after exporting the logs.

❖ **Exploration of the data**

The data use for this paper are those collected from Huawei NCE Tx, it was collected from "2016-03-28 12:24:20" to "2022-09-01 12:22:58".

❖ **Alarm sources**

The image below in Figure 8 gives the top 10 sources of alarms; this information helps the maintenance engineer in the search for the root causes of the problems.

❖ **Alarm severity levels**

Figure 9 presents the command used for classification of alarms based on their severity. In the operation of the network, the knowledge of the severity level allows to define the priorities during the resolution of the problems. The critical severity problems strongly impact the services making them unavailable and are the first ones to be addressed for resolution. We have 4 levels of severity for alarms, they are the following:

- ✓ Minor;
- ✓ Critical;
- ✓ Major;
- ✓ Warning.

The knowledge on the number of occurrence of the alarms is also a form of classification of the alarms, the methods used for the resolutions of the most frequent alarms can be reused, moreover in terms of preventive maintenance; we

can anticipate and prevent certain faults from occurring. Figure 10 presents the top 10 recurring alarms.

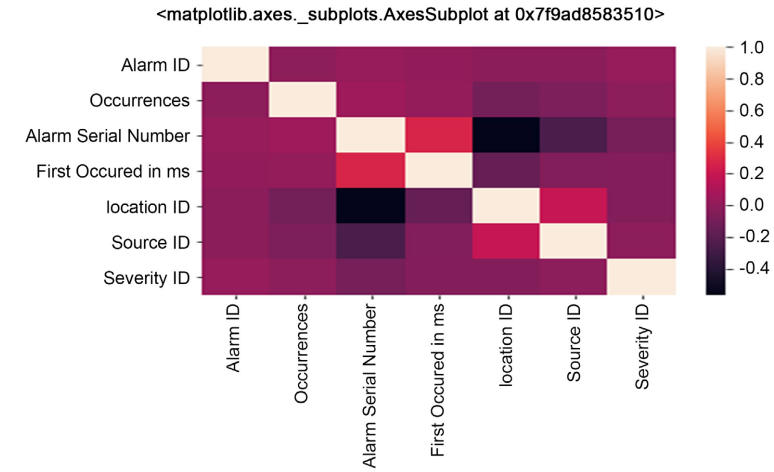
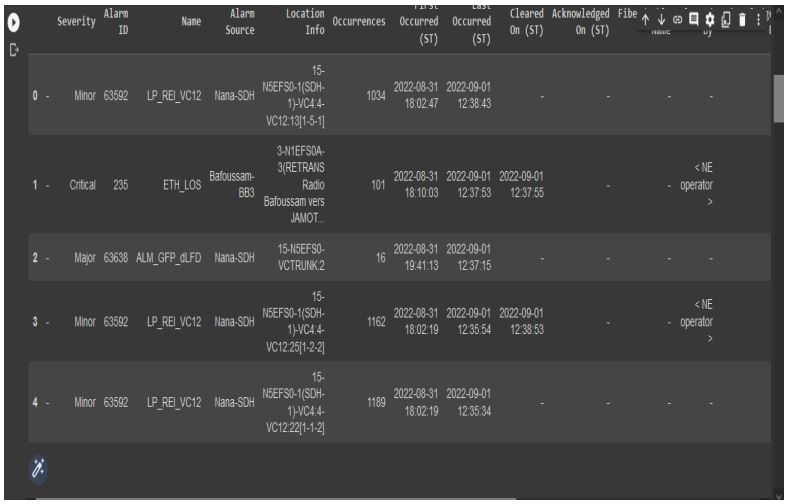


Figure 6. Correlation between the dataset features.



	Severity	Alarm ID	Name	Alarm Source	Location Info	Occurrences	First Occurred (ST)	Last Occurred (ST)	Cleared On (ST)	Acknowledged On (ST)	Fibe
0	Minor	63592	LP_REI_VC12	Nana-SDH	15-N5EFS0-1(SDH-1)-VC4:4-VC12:13[1-5-1]	1034	2022-08-31 18:02:47	2022-09-01 12:38:43	-	-	-
1	Critical	235	ETH_LOS	Bafoussam-BB3	3-N1EFS0A-3(RETRANS Radio Bafoussam vers JAMOT...	101	2022-08-31 18:10:03	2022-09-01 12:37:53	2022-09-01 12:37:55	-	-
2	Major	63638	ALM_GFP_dLFD	Nana-SDH	15-N5EFS0-VCTRUNK:2	16	2022-08-31 19:41:13	2022-09-01 12:37:15	-	-	-
3	Minor	63592	LP_REI_VC12	Nana-SDH	15-N5EFS0-1(SDH-1)-VC4:4-VC12:25[1-2-2]	1162	2022-08-31 18:02:19	2022-09-01 12:35:54	2022-09-01 12:38:53	-	-
4	Minor	63592	LP_REI_VC12	Nana-SDH	15-N5EFS0-1(SDH-1)-VC4:4-VC12:22[1-1-2]	1189	2022-08-31 18:02:19	2022-09-01 12:35:34	-	-	-

Figure 7. Data header.

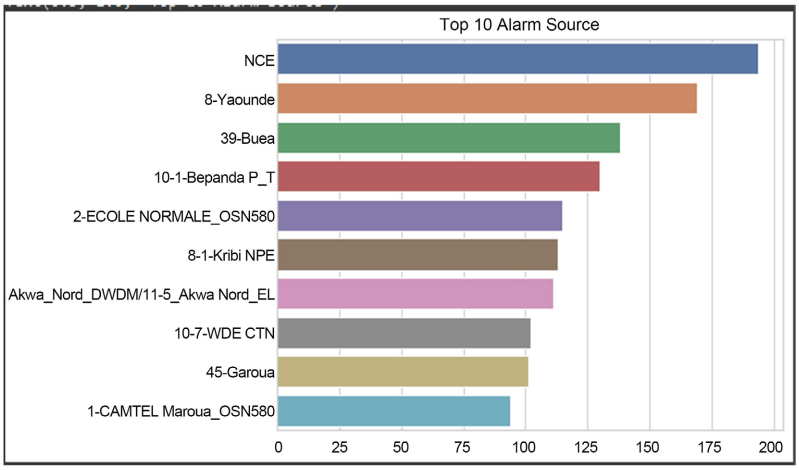


Figure 8. Top 10 sources of alarms (Network entities).

```
[14] df["Severity"].unique()

     array(['Minor', 'Critical', 'Major', 'Warning'], dtype=object)
```

Figure 9. Alarm severity levels.

```
alarms = df.Name.value_counts()
alarms = alarms[:10]
alarms.head(10)

HP_UNEQ         870
R_LOS           766
T_ALOS          536
CHAN_ADD        361
ALS_ACTIVE      277
ETH_LOS         205
TU_AIS          205
AU_AIS          197
DOWN_E1_AIS     189
J0_MM           146
Name: Name, dtype: int64
```

Figure 10. Top 10 recurring alarms.

❖ **The most recurrent alarms in the dataset**

By analyzing the most recurrent alarms in the dataset, it is possible to identify some common points of failure and find the root cause of a problem, it can be fiber cut, power problem or others issues. Figure 10 presents the Python commands used to classify alarms with respect to their occurrence number while Figure 11 presents a graphic with alarms type (name) with respect to the occurrence number.

❖ **The top 10 most recurrent alarms in our dataset**

The knowledge of the most recurrent alarms guides the maintenance engineer in the search for the root causes in the network.

❖ **Top 10 critical alarms**

❖ **Top 10 major alarms**

Note: It can be seen that the top 10 major alarms are similar to the top 10 critical alarms.

Figure 12 and Figure 13 present occurrence of each alarm by severity degree, Figure 12 is related to alarms with a critical severity while Figure 13 presents alarms with major as alarm severity. From the previous paragraph, we have presented different ways of classifying alarms, depending on the source of the alarm, the level of severity of the alarm, the number of occurrences of the alarm and the geographical area affected by the alarm. It is now important to use two machine learning algorithms to establish correlations between these alarms with the ultimate goal of quickly identifying the root causes when problems occur and providing solutions in order to respect the **Services Level Agreement** between the optical network owner and the various customers.

2.3.3. Learning and Creation of Prediction Models

After importing and cleaning the dataset, we need to start applying the different

clustering algorithms for training. The data to be classified being unlabelled data, we are therefore facing a situation of Unsupervised Learning. Also, the algorithms retained for the research of our model are constituted of K-Means and DBSCAN.

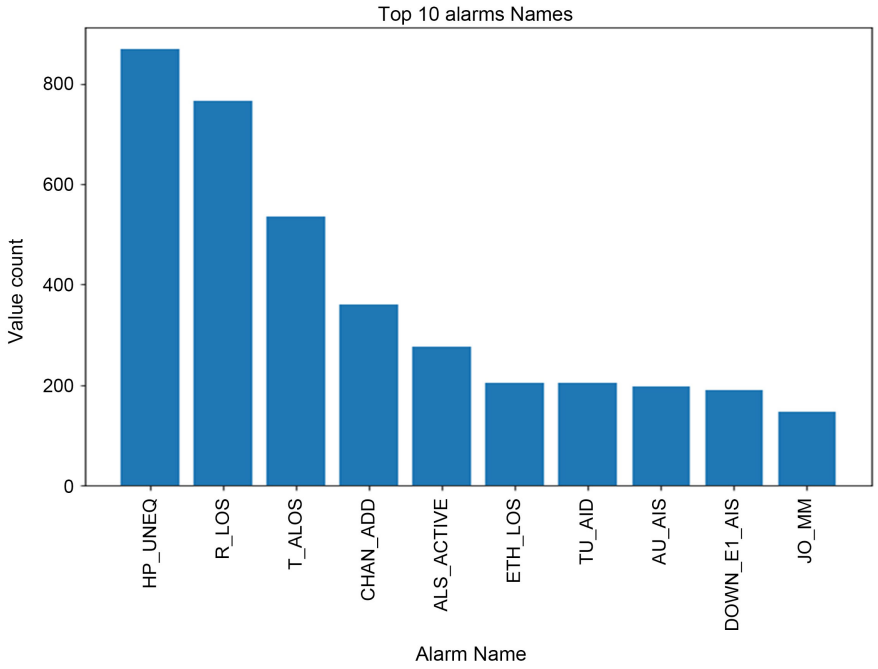


Figure 11. Visualization of the most recurrent alarms.

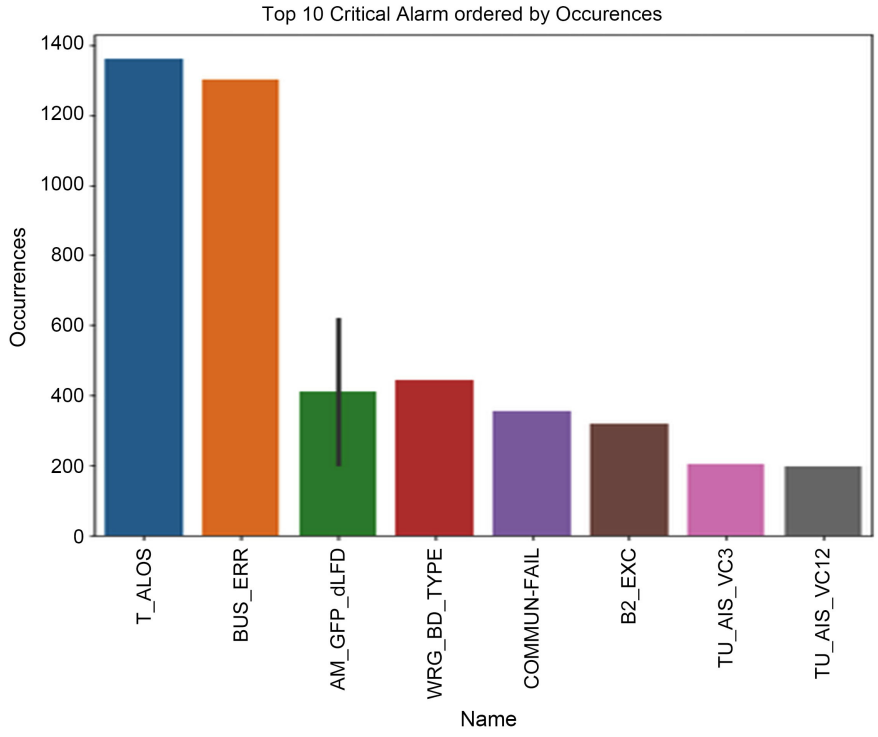


Figure 12. Top 10 critical alarms.

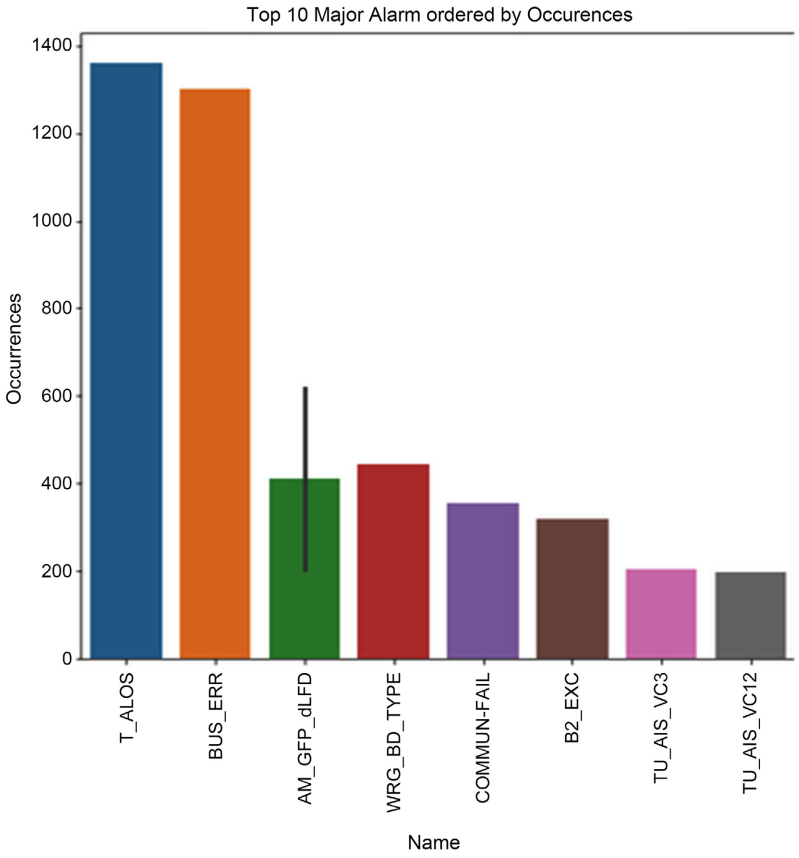


Figure 13. Top 10 major alarms.

Clustering with the K-Means algorithm

We did an 8 and 5 features approach to the dataset to find the underlying causes:

- "Alarm ID",
- "Occurrences",
- "First Occured in ms",
- "Last Occured in ms",
- "Location ID",
- "Source ID",
- "Severity ID",
- "Cleared status".

Choice of the number of clusters

To determine the optimal number of clusters, we use the "elbow" method.

This method reveals that there could be 5 underlying behaviors in the 8-feature datasets. Figure 14 presents the result obtain after implementing the "elbow" function on the dataset and we get the coordinates of the cluster centers this is presented in Figure 15.

The second method reveals that there could be 4 underlying behaviors in the 5-feature datasets (see Figure 16) and we get the coordinates of the cluster centers which are presented in Figure 17.

Clustering with the DBSCAN algorithm

We have this model with the configurations summarized in Table 1.

By using the python code, we obtained the result presented in Table 2.

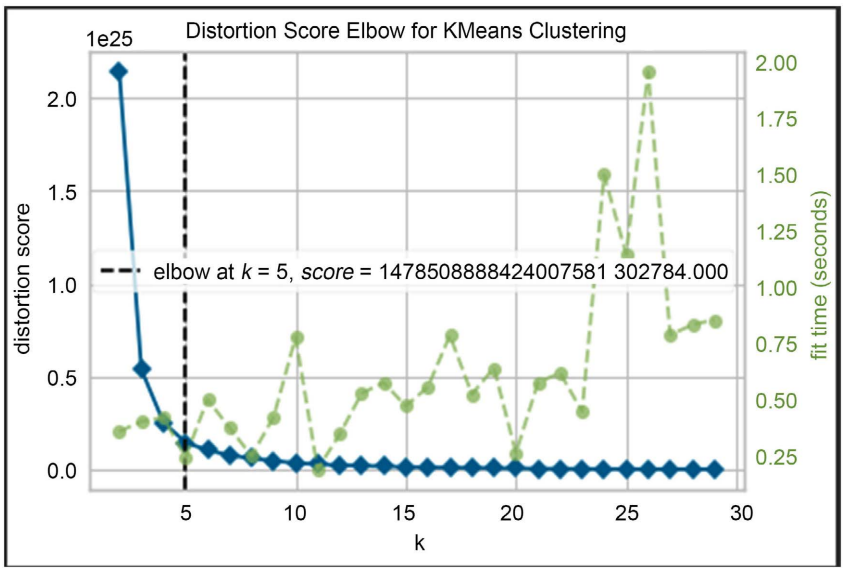


Figure 14. "Elbow" method with K = 5.

	Alarm ID	Occurrences	First Occured in ms	Last Occured in ms	location ID	Source ID	Severity ID	cleared status
0	1.547644e+06	8.603422	1.657511e+12	1.657740e+12	1681.643791	98.584775	1.071319	1.468666e-01
1	9.747590e+02	1.051282	6.438258e+11	6.438270e+11	2485.974359	102.805128	1.297436	1.025641e-02
2	8.010309e+03	1.003697	1.562229e+12	1.562229e+12	3523.556377	167.401109	0.892791	-1.387779e-17
3	6.358716e+03	1.036225	1.619412e+12	1.619838e+12	3057.915157	119.970448	1.408961	-1.387779e-17
4	1.256094e+04	1.004808	1.490818e+12	1.490891e+12	3272.004808	180.211538	1.447115	-1.387779e-17

Figure 15. Coordinates of the centroids with K = 5.

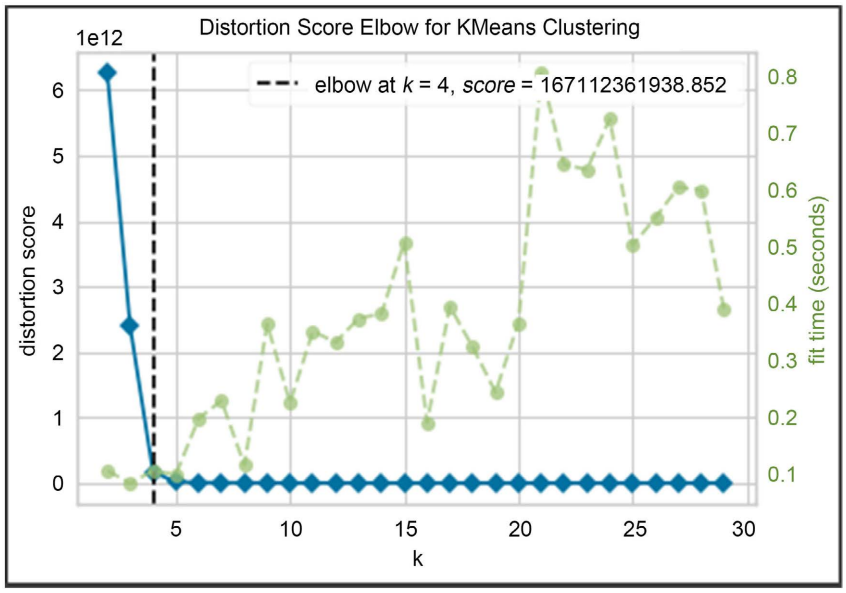


Figure 16. "Elbow" method with K = 4.

	Alarm ID	location ID	Source ID	Severity ID	Occurrences
0	2.338226e+03	2124.307209	112.255215	1.088037	4.611656
1	1.000000e+09	1327.250000	74.000000	1.750000	1.000000
2	5.000020e+05	2360.875000	74.000000	2.000000	2.500000
3	6.360517e+04	1733.294931	81.569892	1.456221	25.626728

Figure 17. Coordinates of the centroids with K = 4.

Table 1. Values to use for the model implementation.

Epsilon	Minpts
1000	100

Table 2. Model results.

Number of data considered as noise	Number of clusters
113	3

To evaluate our two models, we will use the silhouette score. The silhouette coefficient or silhouette score is a measure used to calculate the quality of a clustering technique. Its value is between −1 and 1. To do this, we execute the python codes presented in **Figure 18** below.

It appears from this code that the K-Means score is 0.92 and the DBSCAN score is 0.87.

3. Results and Discussion

3.1. Case of the K-Means Algorithm

After the implementation of the K-Means algorithm on the NCE-Tx dataset, we obtained the results presented below. It will be divided into two parts part A and part B (depending on the number of features used).

3.1.1. Part A—Approach with 8 Features

❖ **Choice of the clusters number**

To determine the optimal number of possible clusters, we opted for the elbow method; the result is presented in **Figure 14**. **Figure 19** presents the coordinates of centroids, we have a total of 5. This method tells us that there could be 5 underlying behaviors in 8-featured datasets. **Figure 20** gives the coordinates of centroids.

❖ Behavior of centroids in the plane formed by Alarm ID and Location ID

Interpretation:

From **Figure 20**, we found five (05) clusters grouping the different alarms of the optical network with similar behavior. Thus:

- The red color for cluster 1;
- Blue color for cluster 2;
- Cyan color for cluster 3;

- Green color for cluster 4;
- Yellow color for cluster 5;
- The black color for our centroids.

And we also observe, that there are noises constituting the elements of our cluster 1.

❖ Behavior of centroids in the plane formed by Alarm ID and Source ID

For the sake of representation in Figure 21, we have limited ourselves to the Alarm IDs from 0 to 300. In this representation, we observe only one noise in the cluster distribution.

```
clustering.labels_
```

```
array([-1, -1, -1, ..., -1, -1, -1])
```

```
db = metrics.silhouette_score(X, clustering.labels_)
km = metrics.silhouette_score(X, y_pred)
```

```
print(km)
```

```
0.9211654090063482
```

```
print(db)
```

```
0.8719933341437353
```

Figure 18. Using the silhouette score.

	Alarm ID	Occurrences	First Occured in ms	Last Occured in ms	location ID	Source ID	Severity ID	cleared status
0	1.547644e+06	8.603422	1.657511e+12	1.657740e+12	1681.643791	98.584775	1.071319	1.468666e-01
1	9.747590e+02	1.051282	6.438258e+11	6.438270e+11	2485.974359	102.805128	1.297436	1.025641e-02
2	8.010309e+03	1.003697	1.562229e+12	1.562229e+12	3523.556377	167.401109	0.892791	-1.387779e-17
3	6.358716e+03	1.036225	1.619412e+12	1.619838e+12	3057.915157	119.970448	1.408961	-1.387779e-17
4	1.256094e+04	1.004808	1.490818e+12	1.490891e+12	3272.004808	180.211538	1.447115	-1.387779e-17

Figure 19. Centroids coordinate after implementation of the algorithm.

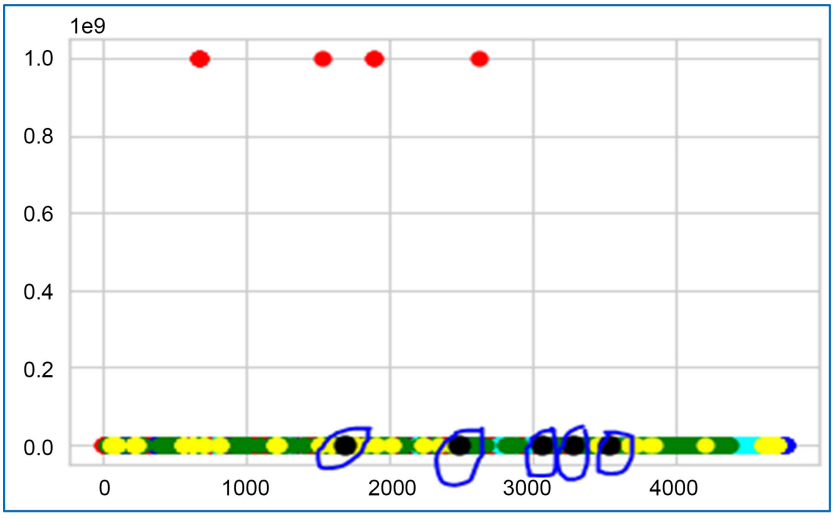


Figure 20. Centroids behavior on the plane Alarm ID and Location ID.

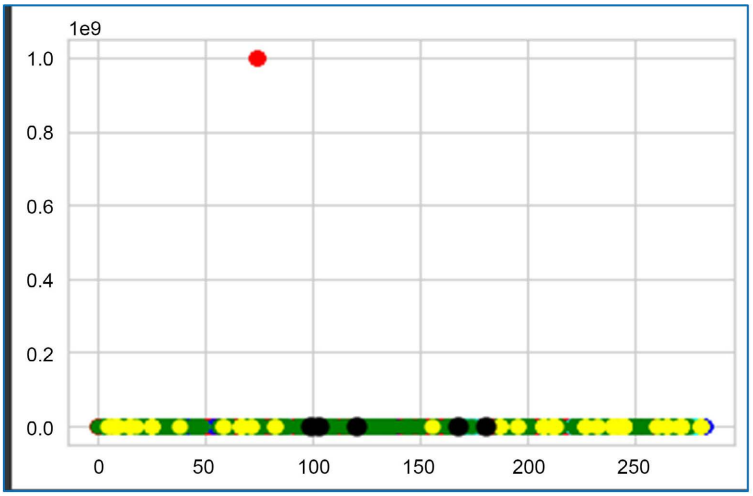


Figure 21. Behavior of centroids in the plane formed by Alarm ID and Source ID.

❖ Visualization

Interpretation: After observing **Figure 22**, we can see that Location ID 70 which corresponds to "5-PQ1-40 (Prepaid Huawei)-PPI:1", is strongly affected by Alarm IDs: 107 and 201, we can see it in **Figure 23**, column 3.

So these behaviors would come from either: Cause 0 called cluster 0 or Cause 3 called cluster 3.

Interpretation: from **Figure 24** according to our distribution,

- We can see that the cause or cluster 0 which occurs frequently is responsible for almost all the alarms recorded.

- Considering the source ID equal to 70 which corresponds to **"11-5-Limbe central"**, the alarms of this source are mostly due to the cause/cluster 1.

Comments: By studying closely the behavior using these 8 features, we can see that the algorithm has clustered the alarms according to time. For example, cluster 0 takes into account the events that occurred from 2022.

3.1.2. Part B—Approach with 5 Features

In this part, we will use the following 5 features:

- "Alarm ID",
- "Location ID",
- "Source ID",
- "Severity ID",
- "Occurrences".

To determine the optimal number of possible clusters, we opted for the elbow method; we have the result presented in **Figure 16**. This method reveals to us that there could be 4 underlying behaviors in 5-feature datasets, see **Figure 26** and **Figure 27**. **Figure 25** gives the centroids coordinates.

Interpretation: From **Figure 27**, with this model the behaviors around location ID 70 are all from cluster 0. The source ID 70 refers to the network entity of Limbe central.

Interpretation: By analyzing **Figure 28** and by considering the source ID

equal to 70 which corresponds to **"11-5-Limbe Central"**, which in fact is the name of the network entity OSN installed in Limbe Central equipment room, the alarms of this source have the same behavior.

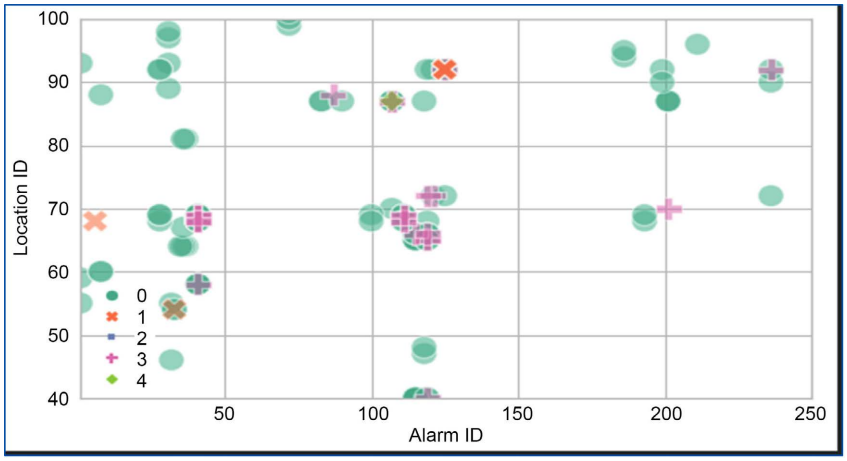


Figure 22. Location ID vs Alarm ID.

	Severity	Alarm ID	Name	Alarm Source	Location Info	Occurrences	First Occurred (ST)	Last Occurred (ST)
76	Major	107	T_ALOS	1-Douala	5-PQ1-40(Prepaid Huawei)-PPI:1	1361	2022-08-31 18:02:47	2022-09-01 12:11:36
5310	Minor	201	DOWN_E1_AIS	1-Douala	5-PQ1-40(Prepaid Huawei)-PPI:1	1	2021-10-28 16:46:29	2021-10-28 16:46:29

Figure 23. Alarm information for ID 76 and 5310.

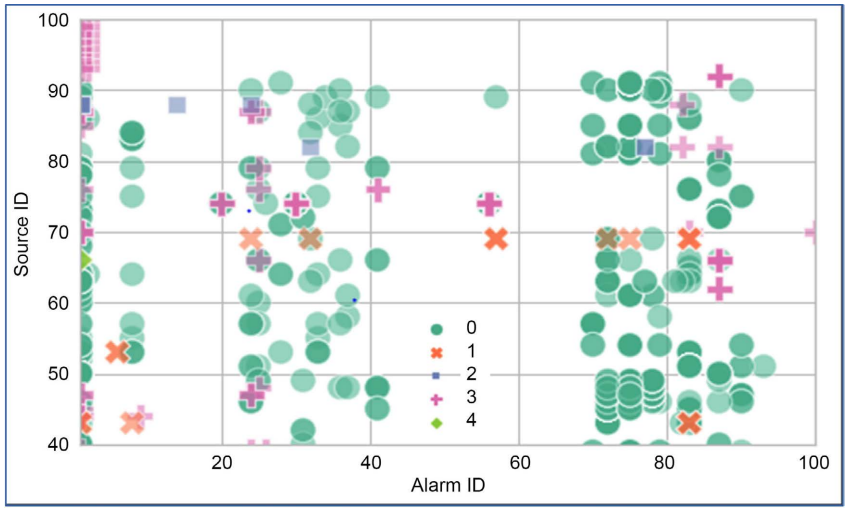


Figure 24. Source ID vs cluster ID.

	Alarm ID	location ID	Source ID	Severity ID	Occurrences
0	2.338226e+03	2124.307209	112.255215	1.088037	4.611656
1	1.000000e+09	1327.250000	74.000000	1.750000	1.000000
2	5.000020e+05	2360.875000	74.000000	2.000000	2.500000
3	6.360517e+04	1733.294931	81.569892	1.456221	25.626728

Figure 25. Centroid coordinates.

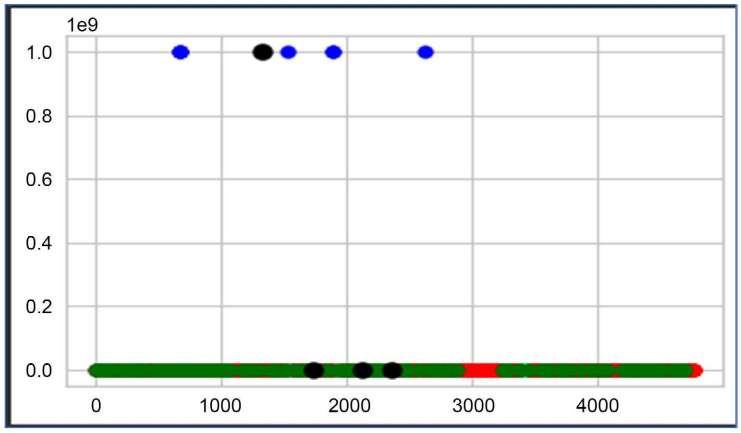


Figure 26. Representation between Alarm ID and Location ID.

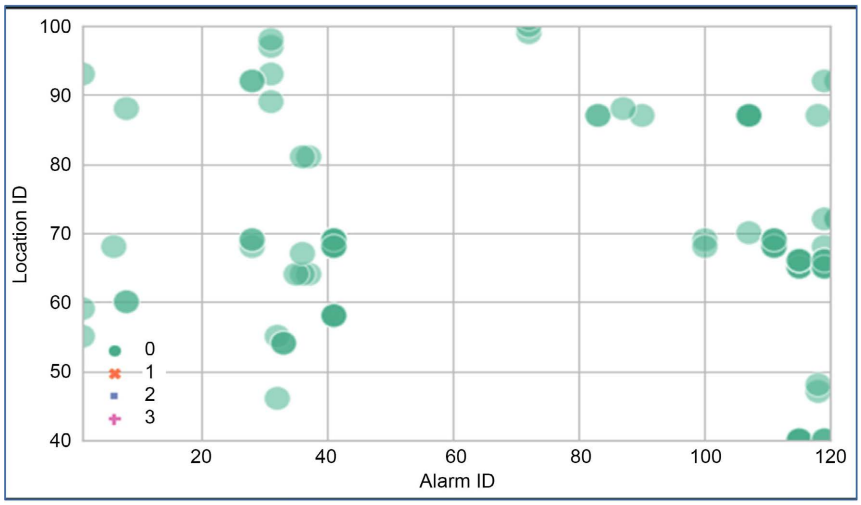


Figure 27. Visualization 3.

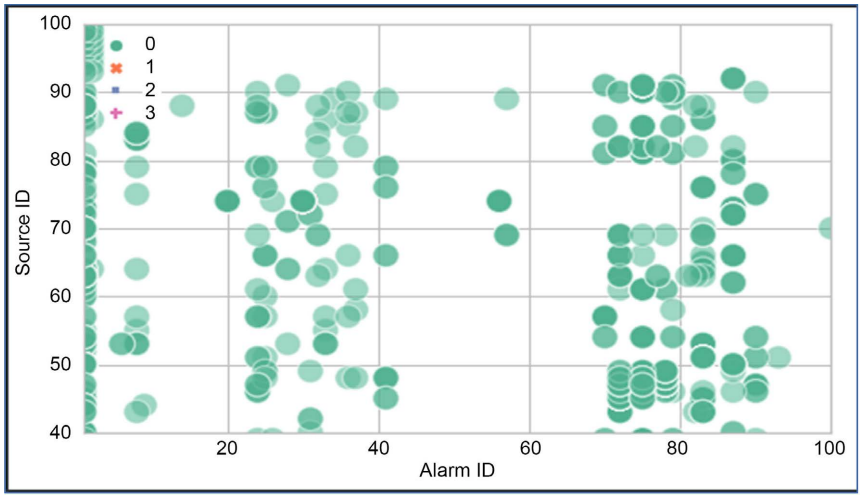


Figure 28. Visualization 4.

3.2. Case of the DBSCAN Algorithm

After implementing the DBSCAN algorithm on the data set from the NCE-Tx,

we obtained the results presented below.

The **optimal** ε is equal to **1000** for a better partitioning of our dataset. This can be seen in the Figure 29 below using the Scikit-Learn library.

❖ Visualization

Interpretation: For the sake of representation in Figure 30, we have limited ourselves to Alarm IDs from 0 to 1000. In this representation, we can see for example that the alarms of the source ID 90 all have the same behavior or are all from the same cluster.

At the end of the experimentation that we have carried out in the previous paragraphs, the machine learning model that we propose is the **K-Means** model. Indeed, this model is retained because its quality of clustering technique is superior to that of **DBSCAN**. Table 3 presents the results of the evaluation of the algorithms.

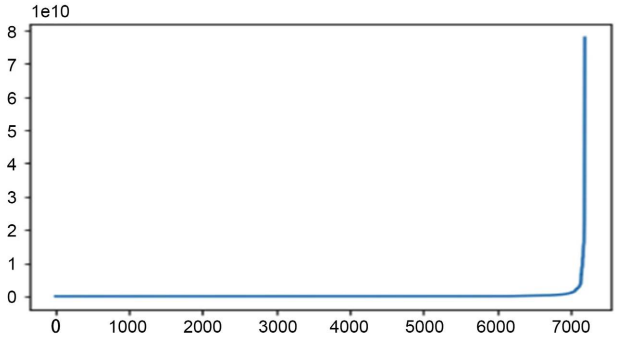


Figure 29. Visualization of the optimal ε.

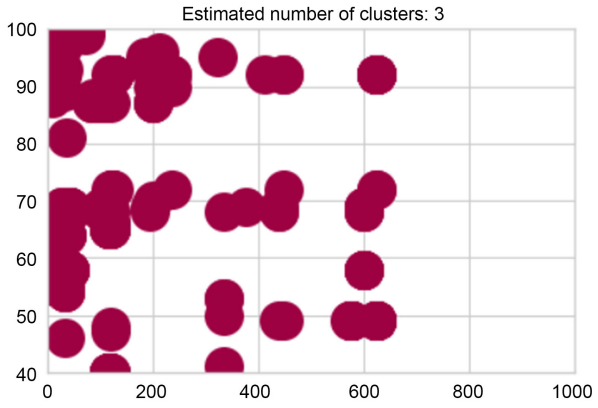


Figure 30. Visualization 5.

Table 3. Results of the evaluation of the algorithms.

Points de comparaison	K-means	DBSCAN
Nombre de clusters	4	3
Variable	K = 4	Epsilon = 1000 Minpts = 100
Nombre de données bruits	/	113
Coefficient de silhouette	**0.92**	**0.87**

For x = 5 features, x being the number of features selected in the log file.

From the classification done using K-Means:

- Cluster 0 is close to the SWDL alarms because it is the alarm that is closest to the first centroid.
- Cluster 1 groups the alarms around the software (expired license, end of free period) because it is the alarm that is closest to the second centroid.
- Cluster 2 is close to the PROTECTION_SUBNET_RISK alarm because it is the alarm closest to the third centroid.
- Cluster 3 is close to the PORT_MODULE_OFFLINE, ALM_GFP, LASER_MOD_ERR_EX, LCAS_PLCT, LCAS_TLCT, LCAS_PLCR, LCAS_FOPR alarms because they are the ones closest to the fourth centroid.

In view of these results, we can deduce the different root causes of the alarms grouped in different clusters using the maintenance guide of the vendor who in this case is Huawei.

The results obtained from our experimentation allow us to determine the root causes of the alarm clusters through the unsupervised classification of the optical network alarms. The unsupervised system proposed solutions for pre-processing the alarms to allow the network administrator to determine the causes in a log file. The work carried out and explored in the state of the art chapter shows us that the exploitation of AI in the maintenance of telecommunication networks has brought great efficiency. However, these works only facilitate the analysis of a log file and do not give the causes of the alarms. With the advent of expert systems, the problem has begun to be addressed, but has been hampered by the fact that these systems become obsolete very quickly in the face of a dynamic and extensive environment. The methodical learning offered by machine learning models therefore brings a step forward towards the intelligent processing of mobile network maintenance work.

4. Conclusions

The present work focused on the classification and correlation of optical network alarms based on Machine Learning algorithms. We applied two Unsupervised Algorithms, on log files of alarms from an optical network operator supervision platform in order to be able to determine the root causes of failures. Our goal was to obtain a machine learning model that will facilitate the analysis of alarm data from the optical layer supervision's platform, record this data and obtain the root causes. This is based on the real alarms of the studied network.

To achieve this goal we have shown the development process of two unsupervised classification algorithms to group alarms from the information present on the log files of the optical network supervision platform.

The results obtained show the interest of the proposed approach and its capacity to facilitate the analysis of log files; this with the aim of obtaining the root causes. These results are therefore globally satisfactory. We propose as perspectives to add a functionality of failure prediction using a supervised model, to implement this model on the network and add in vendor management platform

machine learning based modules even if they are licenses based to help network operators to easily detects root causes of problems and solve them.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Baraketi, S. (2015) Ingénierie des réseaux optiques SDH et WDM et étude multicouche IP/MPLS sur OTN sur DWDM. Réseaux et télécommunicationsm. Universite Toulouse III Paul Sabatier, Français.

[2] Marwa, M.L. and Samira, M.M. (2017) Etude des Reseaux D'Acces Optique Exploitant le Multiplexage en Longueurs D'onde.

[3] Expert. https://www.expert.ai/blog/machine-learning-definition

[4] Talend. https://www.talend.com/fr/resources/what-is-machine-learning

[5] Müller, A.C. and Guido, S. (2016) Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media Inc., Sebastopol, 1-319.

[6] Berthier, E. (2019) Une Introduction au Machine Learning.

[7] Chloé-Agathe (2019) Introduction au Machine Learning. 9, 10.

[8] Teahouse. https://teahouse.fifty-five.com/fr/petit-guide-du-machine-learning-partie-4-lapprentissage-par-renforcement

[9] Deussom, E., Matemtsap, M.B., Tchagna, K.A., *et al.* (2022) Machine Learning-Based Approach for Designing and Implementing a Collaborative Fraud Detection Model through CDR and Traffic Analysis. Transactions on Machine Learning and Artificial Intelligence, **10**, 46-58.

[10] Bernabe, B., *et al.* (2022) Comparing Machine Learning Algorithms for Improving the Maintenance of LTE Networks Based on Alarms Analysis. *Journal of Computer and Communications*, **10**, 125-137. https://doi.org/10.4236/jcc.2022.1012010

[11] Klemettinen, M., Mannila, H. and Toivonen, H. (1999) Rule Discovery in Telecommunication Alarm Data. *Network and Systems Management Journal*, **7**, 395-423. https://doi.org/10.1023/A:1018787815779

[12] White, T. and Ross, N. (1996) Fault Diagnosis and Network Entities in a Next Generation Network Management System. *In Conference Reports: Expert Systems Applications in Artificial Intelligence*, Paris, 1996, 517-522.

[13] Bouillard, A. and Ronot, B. (2013) Alarms Correlation in Telecommunication Networks. https://hal.inria.fr/hal-00838969

[14] Amani, N., *et al.* (2005) A Case-Based Reasoning Method for Alarm Filtering and Correlation in Telecommunication Networks. *Canadian Conference on Electrical and Computer Engineering*, Saskatoon, 1-4 May 2005, 2182-2186. https://ieeexplore.ieee.org/document/1557421

[15] Benzaki, Y. (2018) Tout ce que vous voulez savoir sur l'algorithme K-Means. https://mrmint.fr/algorithme-k-means

[16] Adinugroho, S. and Sari, Y.A. (2018) Implementasi Data Mining Menggunakan Weka. Universitas Brawijaya Press, Kota Malang.

[17] Machine Learning & Clustering: Focus sur l'Algorithme DBSCAN. https://datascientest.com/machine-learning-clustering-dbscan