Extraction and Early Detection of Anomalies in Lightpath SNR using Machine Learning Models

Stéphanie Allogba, Banti Laure M. Yaméogo and Christine Tremblay

***Abstract*— In a context of ever-increasing traffic, a degradation of the optical layer could affect client demands, in particular the quality of service provided by telecommunications operators. Thus, the rapid detection and prediction of performance degradations occurring in the optical lightpath could help to minimize errors in the network. This paper proposes a failure detection model, equivalent to a performance degradation detection model, but based on machine learning (ML) techniques, namely, the interquartile range (IQR) and the support vector machine (SVM) methods. Note that this model is built from performance metrics monitored on real optical lightpaths. In addition, our model can both label the anomalies to be defined on the data and capture the features that will be used. Feature engineering is explored using three ML techniques, namely the Boruta algorithm, the Random Forest classifier and the recursive feature elimination (RFE), to select the most useful features for the implementation of the model. Tested on monitored performance metrics, the validation phase shows that the model using the RFE method gives us the best results with an F1-score and a recall of 99.51% and 100%, respectively. These results prove the model's ability to detect in advance the degradation of the performance of the network.**

***Index Terms*— Anomaly Detection, Time Series Decomposition, Interquartile Range Method, Performance Degradation Detection, Boruta, Feature Engineering, Feature Selection, Random Forest, Recursive Feature Elimination, Random Forest, Statistical Analysis, Support Vector Machine.**

I. Introduction

As optical WDM transmission systems are designed to carry increasingly data-intensive applications over large distances, network failures can lead to increasingly larger losses in data. Network failures can be classified into two groups: hard failures (fiber cuts, equipment failures, etc.) and soft failures (deterioration of equipment performance, aging of fiber and equipment, etc.). Hard failures lead to loss of signal and therefore are service impacting, whereas soft failures lead to degradation of signal quality. Soft failures can be detected by monitoring networks and analyzing the networks' performance metrics [1]. These performance metrics can be used to identify and locate the root causes of network performance degradations and failures. One strategy to limit the impact of performance degradations and to prevent failures is to use system margins. However, these margins can be either too low, which increases the probability of failure and leads to a significant presence of false alarms in the network, or too high, which leads to excessive usage of network resources [1-3]. Another strategy is to use ML techniques for failure prediction or detection in the network [4, 5]. Failure prediction aims to analyze and exploit historical data collected in the network to detect performance trends over a given time horizon to detect degradations before they occur. Failure detection aims to detect or identify the degradation of network performance metrics at an early stage, based on classification algorithms [6].

In this paper, we propose a network failure detection model based on the early detection of anomalies observed in the signal-to-noise ratio (SNR) time series for 8 lightpaths deployed in a production network. The proposed model consists of two modules, namely an anomaly definition module and an anomaly detection module. The first module makes it possible to define and extract features and labels based on the interquartile range (IQR) approach. The second module is an anomaly detection model based on the support vector machine (SVM) algorithm.

The remainder of this paper is organized as follows: Section II defines the problem and presents an overview of previous works. Section III defines the notion of anomaly and describes the anomaly definition module. Section IV.A describes the feature engineering process used to select the best features to use in the SNR anomaly detection module. The implementation details of the anomaly detection module, as well as the performance results, are presented and discussed in Section IV.B. Finally, Section V summarizes the conclusions of this work and highlights the potential of the proposed method for network operators.

II. Problem Definition and Related Work

A. Related work

Wang, et al. [2] proposed a hard failure detection model based on SVM to identify the presence or absence of a failure in the network equipment. The features considered by the authors are defined by the physical parameters of the transceivers (input optical power, output optical power, laser current, laser temperature, and environmental temperature) and the statistical properties (maximum, minimum and mean values) of the bit error rate (BER). Moreover, the labeled failures were defined by using an arbitrary threshold set on the

Manuscript submitted on July 2, 2021.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant RGPIN-2019-03972.

S. Allogba, B. L M. Yaméogo and C. Tremblay are with the Network Technology Lab, Department of Electrical Engineering, École de technologie supérieure, Montréal, Qc, Canada H3C 1K3 (e-mail: christine.tremblay@etsmtl.ca).

unusable time of the equipment. Such thresholds are highly dependent on the expert and might not be representative of observed hard failures in a real context.

Conversely, Vela et al. [7-9] focused their work on soft failures by implementing a finite state machine (FSM) and a Bayesian-network-based algorithm to detect BER anomalies reflected by the unexpected state changes observed in the BER data. The labels defining these BER anomalies were established using a fixed threshold, calculated from the mean and standard deviation of the observed BER data. Moreover, the study used synthetic BER data generated from a 320-km lightpath carried on a lab testbed. The features implemented in their anomaly detection model are the statistical properties of BER data over observation windows ranging from 2 minutes to 10 days.

Shahkarami, et al. [10] compared three different methods, namely SVM, random forest (RF) and neural network (NN) to detect soft failures in the network. The authors defined the failures as BER changes with respect to a threshold (defined by an expert) detected during observation periods ranging from 18 to 73 minutes. The results show that the RF algorithm outperformed the other algorithms in terms of prediction accuracy. Similar to the work in [7-9], the authors tested their models on synthetic BER data from a 380-km lightpath on an experimental testbed using statistical parameters as features.

Chen et al. [1, 11] proposed a NN-based model to detect anomalies in the power data of an optical lightpath. In their study, the anomalies were defined by observing the distortions in the distribution of power values. The authors defined the labels by using a density-based method of clustering data. This method, unlike previous studies, allowed the authors to define anomalies without relying on a fixed threshold. Subsequently, an NN algorithm was applied using the densities resulting from the clustering method as features. Similar to the previous studies, the authors tested their models on synthetic data from 6 lightpaths carried on a 220-km network testbed.

In summary, the characteristics of the ML models proposed thus far for network-failure detection have three aspects in common. The first is that the models were trained and tested using synthetic data. These synthetic data were extracted from lightpaths ranging from 220 to 380 km carried in lab experimental setups. The second is that the anomalies or failures are defined based on a fixed threshold-based approach, except in one case. This being said, due to the random nature of the anomalies and their infrequent occurrence, the first two aspects are not representative of the real contexts of the network. Finally, the third common aspect pertains to the choice of features used in the models. These are mainly the statistical parameters of the observed data (minimum values, maximum values, average values, etc.) gathered during an observation period of only a few hours. However, it should be noted that in [1, 11], the authors used different characteristics for the last two aspects.

B. Detection of Anomalies in SNR Time Series

The best approach to improve the performance of anomaly detection models is to avoid using fixed-threshold methods. The models should instead be based on similarity extraction approaches (unsupervised method) or on variable threshold methods using statistical approaches (supervised method). As well, the anomaly detection models would benefit from being trained using field data, which has a good diversity of features.

In this work, we propose a method for the early detection of performance anomalies based on this approach. The performance metrics are obtained and the proof of concept is demonstrated by using SNR time series for 8 lightpaths deployed in the production network of a large carrier in the United States. The method involves two modules. The first module can automatically define the anomalies observed in the SNR time series. The second module can perform early detection of the SNR anomalies. The models are constructed and tested on field data, which make them better adapted for real-world applications.

Fig. 1 shows the methodology used to implement the soft failure detection model. Each of the four steps is detailed in the following sections. Steps 1 and 2 are part of the anomaly definition module, while steps 3 and 4 are part of the detection module.

The first step (Section III.A) involves data preprocessing, which consists in analyzing and cleaning the collected raw data to extract the features to be implemented in the model. One of the particularities of the model is that it uses as features the data from the time series decomposition method, the collected performance metrics, as well as external factors that can be related to human activities in the central offices (COs).

The second step (Section III.B) consists in defining and extracting the anomalies. Unlike previous studies [2, 7-9], the anomalies (labels to be used in the model) are defined without relying on the fixed threshold that is provided by the operators, by only considering the unique patterns of the anomalies.

The third step (Section IV.A) deals with feature engineering. Its aim is to select the features that have the most impact on the anomalies observed in the SNR time series, while keeping in mind that the features chosen will affect the performance of the ML-based anomaly detection models [6]. The three ML techniques used to select the features are: the Boruta algorithm, RF algorithm and RFE method.

Finally, the last step (Section IV.B) is the implementation of SNR anomaly detection models using the SVM algorithm. This includes the model optimization phase, the model construction phase and the model performance evaluation phase. Four SNR anomaly detection models are presented by different feature sets in order to evaluate the impact of the features on SNR anomalies.

III. Anomaly Definition Module

This section will begin with a presentation of the knowledge base (KB) of the field data used in this work. Then, more details are provided on the anomaly definition module, including the definition of anomalies in the context of our study as well as the extraction procedure and the presentation of the features.

This article has been accepted for publication in IEEE/OSA Journal of Lightwave Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3134098

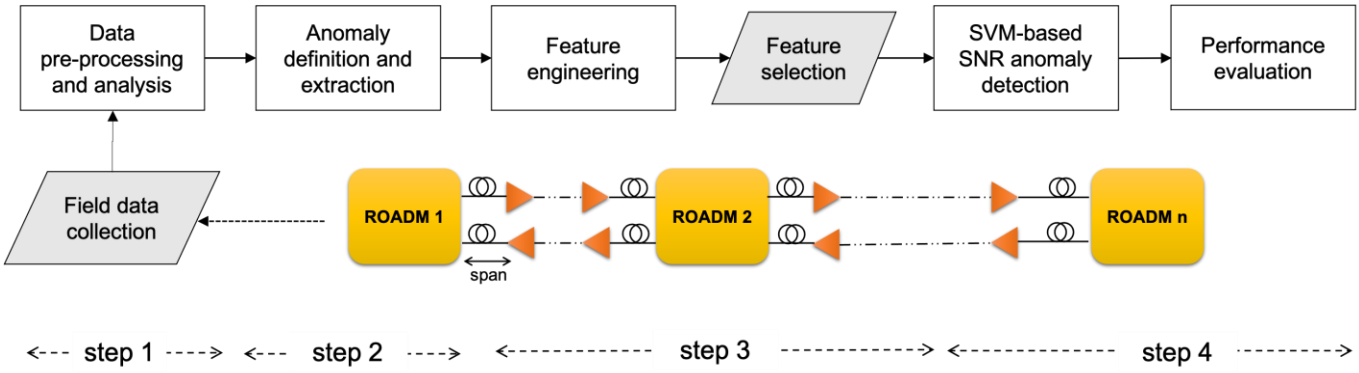


Fig. 1. SNR anomaly detection methodology

A. *KB description*

The knowledge base (KB) used in this work includes performance metrics (PMs) monitored on eight (8) lightpaths carried in the production network of a Service Provider in North America. The lightpaths are deployed on 6 different routes, both buried and aerial fiber plant types, ranging from 100 km (regional) to 1400 km (metropolitan and long haul). Note that the routes are located in different geographical areas. Lightpaths 4 and 8 share a common 1250-km segment of the same 1280-km route and lightpaths 5 and 6 share a common 200-km segment of the same 330-km route.

The links include amplification sites spaced from 25 to 50 km apart and ROADM sites. All the lightpaths are 100 Gb/s polarization multiplexed quadrature phase shift keying (PM-QPSK) channels.

The PMs include received optical power (P_{RX}) and SNR time series collected at 15-minute sampling intervals over an observation period of about 12 months. The resulting KB contains 259,837 instances for each PM. Moreover, there are a number of gaps (282 missing instances at total) in each time series. These gaps, whose duration range from a few minutes to several hours, typically result from software upgrades or maintenance activities. The median value over the observation period is used to fill the gaps in the time series. It is assumed here that selecting a fixed value will not affect the distribution of the data due to the small percentage of missing data for the entire database (0.11%).

B. *Anomaly definition*

The anomaly definition module is divided into two parts. The first part consists of defining and identifying the labels that are to be assigned to the instances in the KB. In other words, it is the definition and the extraction of SNR anomalies.

In general, the notion of anomaly is strongly related to the application domain. Typically, anomalies represent points in a dataset that do not conform to the notion of normal behavior of the domain. Normal behavior, on the other hand, is defined by the interval in which the majority of values lie [12].

In our context, the definition of anomalies was based on the interquartile range (IQR) approach, also known as the *mid-spread* or *boxplot rule*. Note that this approach is widely applied in intrusion detection systems and the computer field [12-14]. Thus, the IQR approach determines the interval corresponding to the normal behavior, thus enabling the observation of anomalies in the data.

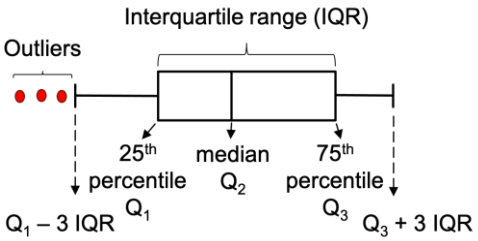


Fig. 2. Illustration of the IQR approach using a boxplot

It consists of estimating threshold values beyond which anomalies are identified based on statistical measures of dispersion. These threshold values are identified by the inner quartile range (25% and 75%) determined from the distribution around the median, as presented in Fig. 2.

In short, anomalies are identified by following two steps. The first step is to calculate the IQR interval, as shown in (1). This interval corresponds to the difference between the 75th percentile and the 25th percentile, also called the upper quartile and the lower quartile, respectively, as follows:

$$IQR = Q_3 - Q_1 \quad (1)$$

where Q_3 is the third quartile or 75th percentile and Q_1 the first quartile or 25th percentile.

The second step is to establish the threshold for defining an anomaly. This is established by expanding the 25/75 baseline by a factor of 3 IQR. Below this baseline, as shown in Fig. 2, any data instance is identified as an anomaly. This is reflected in (2) where any data instance less than Q_1 minus 3 IQR is declared as an anomaly.

$$anomaly \leq Q_1 - 3 \times IQR \quad (2)$$

The *anomalize* package from Rstudio was used to implement the IQR approach in the anomaly definition module [15].

Fig. 3(a) shows the anomalies obtained for the 8 optical lightpaths using the IQR approach and Fig. 3(b) shows an example of anomalies in the SNR data for one of these optical lightpaths. Note that the duration of these anomalies (or abrupt

This article has been accepted for publication in IEEE/OSA Journal of Lightwave Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3134098

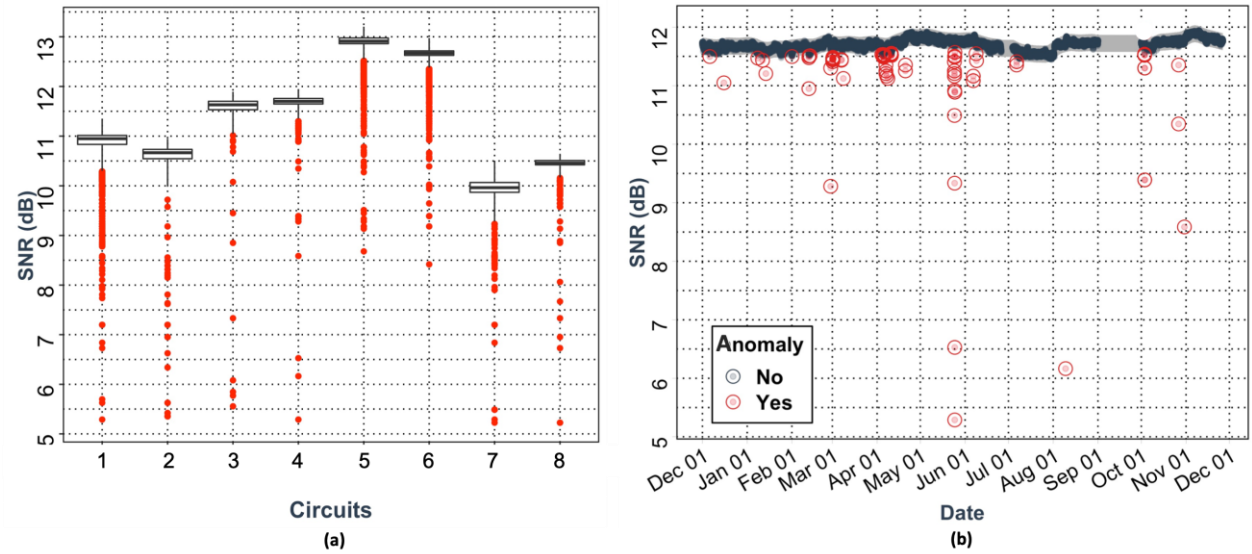


Fig. 3. (a) Boxplots identifying the anomalies in SNR time series for 8 deployed lightpaths using the IQR approach; (b) Example of normal and abnormal data instances observed in the SNR data of one of the 8 optical lightpaths

changes in SNR) is variable and typically range from 15 minutes to less than 2 hours.

Based on the definition and extraction of anomalies, two classes are defined: class 0 (normal behavior or non-anomaly) and class 1 (abnormal behavior or anomaly). Thus, the anomaly definition module extracts 257,739 instances labeled 'non-anomaly' and 2,098 instances labeled 'anomaly' from the database. Note that the 'anomaly' instances represented less than 1% of the database. This strong disproportion of the data could affect the accuracy of the models by overestimating the detection accuracy. To deal with the imbalanced dataset, the number of "non-anomaly" instances has been reduced (in a random fashion) to approximately 4% of the total number of non-anomaly instances, as proposed in [6]. The resulting KB contains 10,490 instances labeled 'non-anomaly' and 2,098 instances labeled 'anomaly'.

C. *Feature identification*

The second part of the anomaly definition module is the identification and extraction of features from the data collected in the network. These features are grouped into three categories.

The first category is derived from the seasonal and trend decomposition (STL) of the SNR data. This decomposition is performed by applying locally weighted regression (Loess) smoothing as described in [16]. Further information on the STL method applied to PM data can be found in [16]. This method consists in separating the time series into three components:

- the trend component (*T*): this represents the general trend in the variations observed in the time series, i.e., an increase, decrease or stability in the changes during an observation period. Note that the trend is not always the same for different slices of the observation period. In other words, for a given time division in the overall observation period, the data may tend to decrease during one slice of that period, while for another time slice, the data may tend to increase or remain stable;
- the seasonal component (*S*): this represents the periodicity observed in the time series, i.e., the patterns that are repeated over a certain period of time (quarter of the year, month, day of week, etc.). Thus, the seasonal component is influenced by seasonal factors;
- the irregular component (*R*): this represents the residual or the remainder of the time series after the other components have been extracted. This component theoretically follows a normal distribution with a zero mean. It can be used to identify random events in the data as mentioned by the authors in [16].

The second category of features is derived from temporal information from the observation period. Indeed, the temporal effects have an impact on the behavior of a time series. The temporal information retained is therefore: the hour (*H*), day of the week (*D*), the number of the week within the year (ind_{week}), the period of the day ($period_{day}$) characterized by the time category [daytime (morning, late morning), evening, night-time].

Note that the extraction of the temporal information (time, day, week, month and year) was extracted from the PM reporting date using the Rstudio lubridate library.

Finally, the last category of features is derived from the performance metrics collected from the network, namely P_{RX}. Moreover, values derived from the SNR data were also added, namely the minimum and maximum values (noted min_{SNR} and max_{SNR}).

IV. Anomaly Detection Module

A. *Feature engineering*

The anomaly definition module identified 10 features, namely: season (S), trend (T), residual (R), minimum and

Table I. Correlation analysis

	S	T	R	min_{SNR}	max_{SNR}	P_{RX}	D	H	ind_{week}	$period_{day}$
S	1									
T	0.2	1								
R	-0.3	-0.2	1							
min_{SNR}	0.9	0.6	-0.3	1						
max_{SNR}	0.9	0.6	-0.3	1	1					
P_{RX}	0.01	0.05	0.07	0.03	0.03	1				
D	-0.005	-0.01	0.002	-0.009	-0.009	-0.005	1			
H	0.001	-0.006	-0.009	-0.002	-0.002	-0.006	-0.01	1		
ind_{week}	0.1	0.5	-0.2	0.3	0.3	0.04	0.002	-0.00002	1	
$period_{day}$	-0.001	-0.02	-0.007	-0.008	-0.008	0.001	-0.02	0.3	-0.009	1

maximum SNR values (min_{SNR} and max_{SNR}, respectively), optical power at the receiver (P_{RX}), day of the week (D), hour (H), number of the week within the year (ind_{week}) and period of the day ($period_{day}$). Note that these features can add either relevant information, noise, or redundant information to the label description [17]. Thus, feature engineering is introduced to understand, analyze and select the features that are to be used in the ML models for SNR anomaly detection.

1) Feature cleaning phase

The first step in the feature engineering is to eliminate interdependent features. Interdependent features only add more noise to the model as they do not provide any additional information on the labels.

Thus, in the feature cleaning phase, a correlation analysis was performed using the Pearson test. The objective of this correlation analysis is to highlight the features that are strongly correlated with each other. Table I shows the results of the correlation study. Three strong correlations were revealed. These correlations are between max_{SNR} and min_{SNR} (correlation value = 1) and between max_{SNR} and min_{SNR} with the season (correlation value = 0.9), and finally between the lower limit and the season (correlation value = 0.9). Note that much weaker correlations were found between the other features.

Subsequently, an arbitrary threshold was set at 0.7 as the dependent correlation value. This threshold value was arbitrarily chosen on the basis that a correlation value above 0.7 is considered strong [18, 19]. Above this threshold, if two features correlate with each other, only one of them is retained and the other is removed from the feature set. Therefore, the minimum and maximum values (min_{SNR} and max_{SNR}) were removed because in addition to correlating with each other, they also correlated with the season.

2) Feature selection

The second step of the feature engineering is to implement the feature selection methods with three ML algorithms that are commonly used in the literature [17, 20, 21], namely the Boruta algorithm, RF algorithm and RFE method. Indeed, the feature selection method consists in removing the least relevant feature for the model. Unlike redundant features that do not provide any additional information to the models, irrelevant features add noise and bias to the models, which can ultimately reduce the performance of the model. Thus, using Python 3's *Scikit-learn* package, the three ML-based feature selection methods were implemented in the model training process to evaluate the relevance of each feature regarding the defined labels.

Boruta

The main objective of Boruta's algorithm is to select the features having the most impact on the desired labels. Here, the desired labels are the anomalies to be detected. The algorithm is implemented following three steps.

The first step is to create a new feature set called *shadow features*. These shadow features are derived from a random combination drawn from the initial feature set. This makes it possible to compare the importance or the score (*Feature importance*, FI) of the features in the initial feature set with that of the shadow features in the next step.

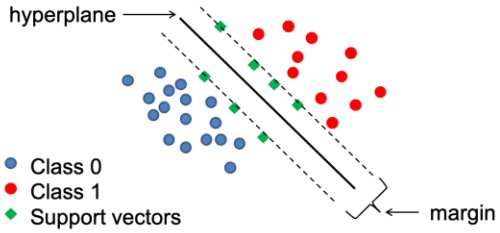
The second step is to create a list called *hit list*. This list contains all features whose FI score is greater than that of the best feature in the shadow features set. To do this, the FI score of the features is first determined from an estimator trained on each newly generated shadow features. The estimator used in our study is the RF estimator. Then, the FI score of each feature in the initial feature set is compared to the maximum value of the shadow features FI score. The feature in the initial feature set having the highest FI score compared to the scores of the shadow features is added to the hit list.

The last step is to repeat the previous two steps until all features are sorted and entered into or rejected from the hit list or until a predefined condition is reached.

Random Forest

The Random Forest algorithm is a learning method based on decision trees. The main idea is to build a tree from the KB composed of features associated with labels. Each node of the tree is a condition on the feature to be inserted into the tree. In other words, it is a condition meant to select optimal features.

Thus, to build the decision tree, first the feature importance measure is determined for each node. This measure is called *impurity*. Features that are relevant for the classification will have a high impurity value, while those that are not associated with the expected labels will have an impurity value near zero.



(a)

	C	G	Kernel
SVM_All	1000	0.1	RBF
SVM_Boruta	1000	0.1	RBF
SVM_RF	1000	1	RBF
SVM_RFE	1000	1	RBF

(b)

Fig. 4. SVM topology: (a) Illustrative example; (b) Optimum parameter set

Table II. Best feature set using feature engineering

Feature selection method	Best feature set
Boruta	S, T, P_{RX}, D, h, ind_{week}
RF	S, T, R, ind_{week}
RF	R

Subsequently, this impurity measure is used as a condition to divide the KB into distinct groups forming the branches of the tree. A tree branch is formed such that the selected features are those that lead to a decrease in the impurity measure. In other words, the features selected at the top of the tree are generally more important than those at the end nodes of the tree.

Recursive Feature Elimination

The objective of the RFE algorithm is to find the smallest set of features for which the model provides good accuracy. It follows three steps. The first is to determine the feature importance measure using an RF estimator trained on the initial feature set. The second step is to eliminate or save the feature. This step consists of pruning the least important features from the initial feature set. A new RF estimator is then trained on the remaining feature set. The third step is to repeat the previous two steps recursively on the pruned set until the desired number of features to be selected is finally reached.

Table II shows the features found by each feature selection method. Thus, for the method using the Boruta algorithm, the best features found are: season, trend, residual, P_{RX}, day, time, ind_{week}. For the method using the RF method, the best features are: season, trend, residual and ind_{week}. For the method using RFE, the best feature is the residual.

Upon completing the feature engineering step, the identified features are fed into the classification model for early detection of SNR anomalies. This results in four SNR anomaly detection models. Each of these models has a different set of features corresponding to the three sets of features found by the feature selection models and to the feature set resulting from the correlation study.

B. Anomaly detection models

The SVM algorithm is used to implement the SNR anomaly detection models. Indeed, it is presented as one of the algorithms frequently used for anomaly detection problems offering a good compromise between computation time and model accuracy [10, 12, 22, 23].

Using this SVM algorithm as well as the various feature configurations found in Section IV.A, four models have been implemented to detect SNR anomalies. These are identified here as SVM_all, SVM_Boruta, SVM_RF and SVM_RFE corresponding to the model using the feature set after the cleaning phase, the Boruta feature set, the RF feature set and the RFE feature set, respectively.

This section includes the presentation of the anomaly detection models and it is divided into two parts. The first part describes the implementation of the SVM algorithm and its optimization. The second part is about the performance evaluation of the models.

1) SVM design for SNR anomaly detection models

The SVM algorithm is mainly used in classification problems and its objective is to distinguish the classes of the database using a hyperplane. This hyperplane acts as a boundary between the classes. In our case, we have two classes: class 0 (anomaly) and class 1 (no anomaly), as defined in Section III.B. The SVM algorithm is described in Fig. 4(a).

The hyperplane can be a linear or nonlinear boundary. It is constructed to maximize the distance between the two classes. This distance between the classes is called the margin. The location of the margin is determined using a subset of observations taken from the training dataset. This subset of observations is called support vectors.

Note that the SVM algorithm is defined by several parameters, namely the regulation or C parameter, the gamma or G parameter and the kernel. The C parameter is used to avoid or reduce the misclassification of each training dataset. The higher the value of C is, the smaller the margin is, yielding a stricter classification. Conversely, the smaller the value of C is, the larger the margin is, giving a broader classification even if some data is misclassified. The G parameter defines the influence of the training points on the calculation of the hyperplane. In other words, a low G parameter means that the points defining the support vectors could be far from the hyperplane. In the same way, a high G parameter means that only the points close to the hyperplane are considered in defining the support vectors.

The kernel represents the function that defines the set of features. It is used to solve the hyperplane equation.

So, in order to implement the four SNR anomaly detection models, the KB is divided according to the ratio 80/20. This ratio corresponds to the training and test datasets.

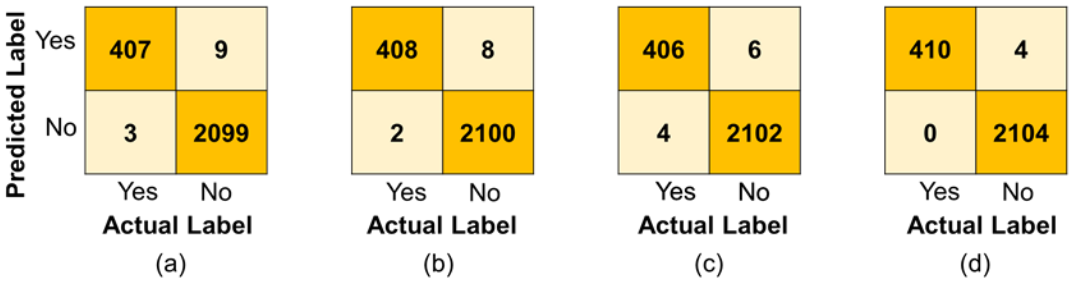


Fig. 5. Confusion matrices for (a) SVM_All model; (b) SVM_Boruta model; (c) SVM_RF model; (d) SVM_RFE model

Table III. Performance metrics of anomaly detection models

	F1 score (%)	Recall (%)	Computation time (ms)*
SVM_All	98.53	99.26	46.87
SVM_Boruta	98.78	99.51	46.87
SVM_RF	98.77	99.02	15.62
SVM_RFE	99.51	100	5

** The models were executed on a system with an Intel® Core™ i5-7200U 2.5 GHz CPU, 8 GB RAM.*

The training dataset is used to optimize the hyper-parameters that are to be used in the models and to train the models. For the optimization phase, the GridSearchCV method of the Scikit-learn library in Python 3 was used. This method consists in performing an exhaustive search on the combinations of hyper-parameter values to find the best combination to use in the models. As for the kernel, it has been fixed. Indeed, because the relationship between the anomalies and the features is nonlinear, it was set to the radial basis function (RBF) [24, 25]. Fig. 4(b) presents the optimal hyper-parameters found with the GridSearchCV method. The values set to be varied for the optimization step are (0.1; 1; 10; 100; 1000) and (1; 0.1; 0.01; 0.001; 0.0001) for C and G, respectively. Thus, the final combinations (C; G) obtained are (1000; 0.1) for the models SVM_all and SVM_Boruta, and (1000; 1) for the models SVM_RF model and SVM_RFE.

A weighting has been applied on each label in the models to address the problem of class imbalance.

2) Evaluation of the SNR anomaly detection models

The models were evaluated using the test dataset corresponding to 20% of the KB. This includes 2,108 instances of class 0 (non-anomaly) and 410 instances of class 1 (anomaly).

Fig. 5 shows the confusion matrices obtained for each model. These provide insight into the accuracy of the models based on the true positive and true negative rates located along the diagonal matrices, as well as the false negative rate. The first two rates, located along the diagonal matrices, are the ratio of instances correctly predicted as anomalies or non-anomalies. In other words, these rates correspond to the correct predictions made by the models on the 2,518 instances of the test dataset. The higher these are, the better the model performs. As for the false positive rate, it indicates the cases where the model predicts a non-anomaly (class 0) while the actual data is an anomaly (class 1). Contrary to the true positive and true negative rates, the lower the false positive rate is, the better the model performs. With its lowest false positive rate (0) and highest true positive rate (410), the SVM_RFE model is the best model, followed by the SVM_Boruta model.

The confusion matrices are also used to determine the model's performance metrics, namely F1-scores and recall, used to assess the models' accuracy. Indeed, the recall quantifies the proportion of correctly predicted anomalies among all the anomalies observed.

The F1-score is a combination of the recall and the precision, where the precision represents the proportion of correctly predicted anomalies among all predicted anomalies.

Note that both the F1-score and recall are robust metrics to use with imbalanced datasets.

The computation time (defined here as the time to run the anomaly detection models) is also evaluated in order to compare the complexity of the models as a function of their execution time. These performance metrics are presented in Table III.

Thus, the F1-score is indicative of the quality of the model. The higher the F1-score is, the greater the model's performance. In Table III, the best performance was obtained with the SVM_RFE model. This model corresponds to the model with the smallest number of features. In other words, it represents the best model to correctly predict SNR anomalies. Indeed, the F1-score decreases from 99.51%, with the SVM_RFE model, to 98.78%, 98.77% and 98.53% with the SVM_Boruta (7 features), SVM_RF (4 features) and SVM_All (8 features) models, respectively. Also note that the smaller the number of features, the better the performance.

As presented in [17], the feature engineering methods improve the performance of the ML models by reducing irrelevant features that can generate errors. This shows that models with fewer features perform better than those with more features.

On the other hand, a good anomaly detection model means low false positive rate in confusion matrices, as shown in Fig. 5.

This is equivalent to high recall. Indeed, the false positive rate represents the cases where the model indicates an instance as non-anomaly (class 0) while the true class of the instance is labeled as an anomaly (class 1). Thus, when analyzing the recall presented in Table III, the best model is SVM_RFE with a recall of 100%. This means that the SVM_RFE model correctly detects all the anomalies present in the test dataset. For the other models, the SVM_Boruta model gives the best

Table IV. Comparative analysis of anomaly detection models

Method	Characteristics			Knowledge base			Accuracy (%)
	Anomaly definition	Features	Anomaly detection	PMs	Lightpath information	Observation (days)	
1 [7-9]	Fixed threshold	Statistical parameters	FSM-Bayesian	BER (lab)	1 lightpath 320 km	60	90
2 [10]	Fixed threshold	Statistical parameters	NN, SVM, RF	BER (lab)	1 lightpath 380 km	24	99.1 (RF)
3 [1, 11]	Clustering	Density	NN	Power (lab)	6 lightpaths 220 km	-	94.8
SVM-RFE	IQR	Residual	SVM	SNR (field)	8 lightpaths 150-1250 km	365	99.5

recall with 99.51%, followed by the SVM_All model with 99.26% and the SVM_RF model with 99.02%.

Finally, with regards to the computation time, the number of features affects the speed of the model. The less features the model uses, the faster it is. The computation time also explains the complexity of the model. In other words, the lower the number of features, the less complex the model is. Thus, SVM_RFE is the least complex model in its execution with the smallest computation time.

Moreover, by comparing the feature found in each of the models, the feature common to all models is the residual. This validates the hypothesis put forward in [16], in which the residual is the best feature to characterize SNR anomalies.

Table IV shows a comparative performance analysis between our best proposed anomaly detection model, namely the SVM_RFE model, and methods found in the literature. Thus, our proposed model is positioned as the best model with the best F1-score, namely 99.51% compared to 90%, 99.1% and 98.4% for methods 1, 2 and 3 published in the literature.

Moreover, the SVM_RFE model uses real field data from a real production network with optical lightpaths of up to 1,250 km, unlike other methods that use synthetic data. Feature engineering was also performed. It extracted the residual as being the best input to use in the SVM algorithm, compared to the statistical values used in methods 1 and 2.

V. Conclusions

In this work, we proposed a new model for the early detection of SNR anomalies. More specifically, the proposed model examines the SNR instances one by one and extracts the best feature from each. Subsequently, the SVM algorithm classifies this new instance, determining whether or not it is an anomaly. This model uses two distinct modules, which are described below.

The first module offers a new definition of SNR anomalies based on the IQR approach. This approach identifies, within the SNR time series of 8 optical lightpaths of a production network, data that could be considered as anomalies or not. This approach differs from the traditional (and often more costly for operators) method, which consists of setting a threshold beyond which the anomaly is detected. In addition, this module selects features different from those proposed in the literature, which are mainly composed of statistical parameters. The features for our approach were grouped into three categories, namely the collected performance metrics, the temporal information and the statistical parameters of the SNR.

The second module is used to detect anomalies in the SNR data. Through feature engineering, it first reduced the list of features to those that are the most relevant by way of a correlation analysis to remove redundant features. Then, three ML algorithms for feature selection (RF, Boruta and RFE methods) were implemented. This reduced the initial feature set to 4, 7 and 1 feature for the RF, Boruta and RFE method, respectively. From these selected features, four anomaly detection models using the SVM algorithm were proposed. The best results were obtained with the SVM-RFE model (1 feature) with an F1-score of 99.51% and a recall of 100%.

In addition, by analyzing the performances of the different models, our approach also presents the residual as the best feature to describe SNR anomalies. The SVM_RFE model, using the residual as feature, appears as a promising solution that could be used in real time to detect SNR anomalies. This model could be implemented in proactive maintenance tools in the event of a network failure due to degraded performance.

Further work will focus on the optimization of the ML models using additional input data to search for complex patterns and periodicities in the SNR data. Moreover, another classification algorithm could be used as a comparison with the SVM method to find the best model to detect anomalies.

References

[1] X. Chen, B. Li, R. Proietti, Z. Zhu, and S. J. B. Yoo, "Self-Taught Anomaly Detection With Hybrid Unsupervised/Supervised Machine Learning in Optical Networks," *Journal of Lightwave Technology*, vol. 37, pp. 1742-1749, 2019.

[2] Z. Wang, M. Zhang, D. Wang, C. Song, M. Liu, J. Li, *et al.*, "Failure prediction using machine learning and time series in optical network," *Optics Express*, vol. 25, pp. 18553-18565, 2017/08/07 2017.

[3] Y. Pointurier, "Design of Low-Margin Optical Networks," *Journal of Optical Communications and Networking*, vol. 9, pp. A9-A17, 2017/01/01 2017.

[4] Francesco Musumeci, Cristina Rottondi, Avishek Nag, Irene Macaluso, Darko Zibar, Marco Ruffini, *et al.*, "A Survey on Application of Machine LearningTechniques in Optical Networks," 2018.

[5] R. Gu, Z. Yang, and Y. Ji, "Machine learning for intelligent optical networks: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 157, p. 102576, 2020/05/01/ 2020.

[6] C. Zhang, D. Wang, L. Wang, J. Song, S. Liu, J. Li, *et al.*, "Temporal data-driven failure prognostics using BiGRU for optical networks,"

IEEE/OSA Journal of Optical Communications and Networking, vol. 12, pp. 277-287, 2020.

[7] A. P. Vela, M. Ruiz, F. Fresi, N. Sambo, F. Cugini, G. Meloni, *et al.*, "BER Degradation Detection and Failure Identification in Elastic Optical Networks," *Journal of Lightwave Technology*, vol. 35, pp. 4595-4604, 2017.

[8] A. P. Vela, M. Ruiz, F. Fresi, N. Sambo, F. Cugini, L. Velasco, *et al.*, "Early Pre-FEC BER Degradation Detection to Meet Committed QoS," in *Optical Fiber Communication Conference*, Los Angeles, California, 2017, p. W4F.3.

[9] A. P. Vela, M. Ruiz, F. Cugini, and L. Velasco, "Combining a machine learning and optimization for early pre-FEC BER degradation to meet committed QoS," in *2017 19th International Conference on Transparent Optical Networks (ICTON)*, 2017, pp. 1-4.

[10] S. Shahkarami, F. Musumeci, F. Cugini, and M. Tornatore, "Machine-Learning-Based Soft-Failure Detection and Identification in Optical Networks," *Journal of Optical Communications and Networking*, 2018.

[11] X. Chen, B. Li, M. Shamsabardeh, R. Proietti, Z. Zhu, and S. J. B. Yoo, "On Real-Time and Self-Taught Anomaly Detection in Optical Networks Using Hybrid Unsupervised/Supervised Learning," in *2018 European Conference on Optical Communication (ECOC)*, 2018, pp. 1-3.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, p. Article 15, 2009.

[13] O. Vallis, J. Hochenbaum, and A. Kejariwal, "A novel technique for long-term anomaly detection in the cloud," in *6th {USENIX} workshop on hot topics in cloud computing (HotCloud 14)*, 2014.

[14] H. P. Vinutha, B. Poornima, and B. M. Sagar, "Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset," Singapore, 2018, pp. 511-518.

[15] M. Dancho and D. Vaughan. (2018). *Package 'anomalize'*. Available: https://github.com/business-science/anomalize

[16] B. L. M. Yaméogo, D. W. Charlton, D. Doucet, C. Desrosiers, M. O. Sullivan, and C. Tremblay, "Trends in Optical Span Loss Detected Using the Time Series Decomposition Method," *Journal of Lightwave Technology*, vol. 38, pp. 5026-5035, 2020.

[17] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, pp. 16-28, 2014/01/01/ 2014.

[18] D. S. Moore, W. Notz, and M. A. Fligner, *The Basic Practice of Statistics*: W.H. Freeman and Company, 2013.

[19] B. Ratner, *Statistical and Machine-Learning Data Mining, Third Edition: Techniques for Better Predictive Modeling and Analysis of Big Data, Third Edition*: Chapman & Hall/CRC, 2017.

[20] U. Stańczyk and L. C. Jain. (2015). *Feature selection for data and pattern recognition*.

[21] F. Degenhardt, S. Seifert, and S. Szymczak, "Evaluation of variable selection methods for random forests and omics data sets," *Briefings in Bioinformatics*, vol. 20, pp. 492-503, 2017.

[22] M. Braei and S. Wagner, "Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art," *ArXiv*, vol. abs/2004.00433, 2020.

[23] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 1310-1315.

[24] C.-W. Hsu, C.-C. Chang, and C. Lin, "A Practical Guide to Support Vector Classication," 2008.

[25] A. Shmilovici, "Support Vector Machines," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds., ed Boston, MA: Springer US, 2005, pp. 257-276.

Stéphanie Allogba received her B.Eng. degree in telecommunications from ESME Sudria, France, in 2011 and her M.A.Sc. degree in telecommunications networks engineering from École de technologie supérieure, Montréal, Canada, in 2015. Her master's thesis focused on control and management algorithms for femtocell wireless networks. In 2021, she received her Ph.D. degree at École de technologie supérieure, with the Network Technology Lab. Her Ph.D. activities and research interests include the application of machine learning algorithms for lightpath classification and performance prediction in optical networks. She is currently working as a research assistant with the Network Technology Lab at École de technologie supérieure.

Banti Laure M. Yaméogo received her B.Eng. degree in system and network management from the University of Aube Nouvelle, Burkina Faso, in 2005 her M.Eng. degree and her Ph.D. degree from École de technologie supérieure, Montréal, Canada, in 2015 and 2021, respectively. Her master's project focused on the development of a simulation model for 100 Gb/s coherent optical links. Her Ph.D. activities and research interests include the statistical analysis and characterization of performance monitoring datasets collected in a production network. She is currently working as research assistant with the Network Technology Lab at École de technologie supérieure.

Christine Tremblay received her B.Sc. degree in Engineering Physics from Université Laval, Quebec City, Canada, in 1984, M.Sc. degree (Energy) from INRS-Énergie, Varennes, Canada, in 1985 and Ph.D. degree (Optoelectronics) from École Polytechnique de Montréal, Canada, in 1992. She is a Full Professor with the Department of Electrical Engineering and Associate Director for the Ph.D. Program at École de technologie supérieure.

She is the Founding Researcher and Head of the Network Technology Lab. Before joining ÉTS, she was a Research Scientist with the National Optics Institute (INO) where she conducted research on integrated optical devices for communication and sensing applications. She has held senior R&D and technology management positions for several organizations. As Engineering Manager at EXFO and Director of Engineering at Roctest, she was responsible for the development of fiber-optic test equipment. She also served as Product Manager at Nortel for DWDM systems. Her research interests include machine learning for optical networking applications, performance monitoring, as well as filterless optical networking and novel optical network architectures. She has been co-instructor for SC314 and SC210 hands-on courses of the Optical Society of America on optical fiber and polarization measurements.

Dr. Tremblay is a Senior member of the IEEE, as well as a member of the Optical Society of America (OSA) and the STARaCom and COPL Strategic Clusters of Quebec FRQNT. She is currently serving as Technical Program Committee (TPC) Member for OFC N3 Subcommittee "Architecture and software-defined control for metro and core networks".