Machine Learning for RF Impairment Classification

A Technical Paper prepared for SCTE by

David Virag
Distinguished System Engineer
Commscope
Suwanne GA, 30024
david.virag@commscope.com

Santhana Chari, PhD
Sr. Director, System Engineer
Commscope
Suwanne GA, 30024
Santhana.Chari @commscope.com

Stephen Kraiman
Distinguished System Engineer
Commscope
Horsham PA, 19044
Stephen.kraiman@commscope.com

Table of Contents

List of Figures

List of Tables

1. Introduction

There has been an explosion in the ability to organize and process large amounts of data over the past 20 years. The introduction of the world wide web has introduced key services such as search engines and social media where massive amounts of data are necessary to sort, filter, and aggregate in ways that enhance the value of the services to their customers. To support the new services that now aggregate massive amounts of data - new tools, algorithms, and pipeline architectures have been developed and have become readily available to attack problems in new disciplines where large amounts of data are now becoming available.

Simultaneously, over the past decade, CableLabs DOCSIS standards have introduced a variety of Proactive Network Maintenance (PNM) tests for the collection of operational data from various network elements such as Cable Modems (CMs) and Cable Modem Termination Systems (CMTS). The operational data available in the network can be extremely large when taken at time intervals sufficient to allow pro-active network management. Historically, these sort of data available in the cable network required one skilled in the domain of radio frequency engineering to interpret the spectral data available from the system. Discerning the quality of an RF spectral signal visually is time consuming and fraught with nuance such that human review of this data is done in a reactive manner where issues are already known as opposed to a proactive manner to determine where issues are arising that may not yet be having major impacts on network performance and quality of service.

However, this data has not been sufficiently used to automate the detection of network impairments and to take remediation measures to alleviate network issues that can result in deterioration of the quality-of-service experience by the customers. Recent advances in Machine Learning techniques to analyze large quantities of data coupled with higher performing hardware computer algorithms have made it possible to evaluate PNM data in near real-time for the purposes of classification of the channel quality and identification of impairments.

This paper explores the use of various Machine Learning algorithms to categorize downstream Full-Band Spectrum (FBS) capture data extracted from the CMs. In this evaluation, all RF impairments are lumped into a single group where the initial characterization of the group is based on a few simple RF evaluation metrics. Various pre-processing techniques to normalize the spectrum data and other challenges that are encountered in the processing pipeline are presented. Different ML algorithms with various levels of complexity were evaluated to identify and categorize the presence of spectrum impairments. Results of our experiments based on real field data collections will be presented. Techniques for closed-loop identification of RF impairments are presented.

2. Challenges of using Machine Learning models for Spectrum Identification

The application of machine learning on spectral samples, while not new, has specific challenges when applied for classification. For supervisory based learning systems, training data must be available to train the algorithm what anomalous spectral data may look like. For spectral based data there is unfortunately no systematic method for obtaining labeled data. Assuming that enough spectral samples are available with associated labels, not all systems include spectral energy across the entire Full-Band System (FBS) capture. A full-band system from center frequencies 93 MHz to 993 MHz contains 151 6-MHz channels. In some cases, only a fraction of the 151 channels are configured for services and include transmitted spectral energy. With only a small number of service channels, issues such as RF Tilt or spectral ripple may be difficult to observe due to the large gaps between regions with transmitted channel energy. These gaps appear as noise in the full-band spectrum.

Another issue is that consumer premise equipment (CPE) necessary to capture the FBS data is not always available. CPE may be occasionally powered off, either by the customer or due to loss of commercial power services. For downstream data, the CPE is necessary to capture the received data spectrum. For this study, CPE were chosen randomly for the dataset.

When using a supervised algorithm, labeling of data can be difficult and costly. For RF spectrum, labeling data may take subject matter experts to meticulously look through samples by manually adding labels to those subjectively determined to be abnormal. Lack of precise boundary definitions can confuse machine learning algorithms and decrease the performance of the ML algorithms.

High volume of data associated with a full-band spectrum requires large memory and storage capabilities. In this study, we pulled more than 15,000 random FBS records for evaluation. This size of data set requires server quality hardware. For example, this data set was unable to process on a laptop computer with 8 GB of memory under the current process implementation.

3. Prior Work

Much of the prior work done for ML for spectrum analysis is based on the use case of spectrum sensing, i.e., identification of whether a spectral block is available for use. Other related work has looked at Time Series Classification using Deep Learning [1]. Some work has been done in more specific cases for DOCSIS and OFDM. [2] discusses the sources of data available in a DOCSIS network useful for machine learning approaches and the application of Deep Learning for classification while [3] evaluates the use of convolutional neural networks (CNN) as a multi-label classifier with RxMer data in an OFDM channel. This paper will focus on downstream FBS data across the entire plant including SCQAM and OFDM spectrum.

4. A Machine Learning Evaluation Pipeline

Data is collected from two CMTSs using the PNM Downstream Spectrum Capture using Simple Network Management Protocol (SNMP) to setup and trigger each data sample. For each CMTS, data was captured from each available cable modem across all service groups at the same time of day once or twice per day and placed into a Cassandra database. The full spectrum captures include FBS data from 93-993 MHz center-frequency channels sampled at 256 bins per channel representing 151 6-MHz data, video, or unused channels.

The database may be queried by specific CMTS source, date and time of capture, interface index, or specific cable modem MAC address. The database includes over two million capture samples. Once the data has been ingested into the database, the general process flow is shown in Figure 1.

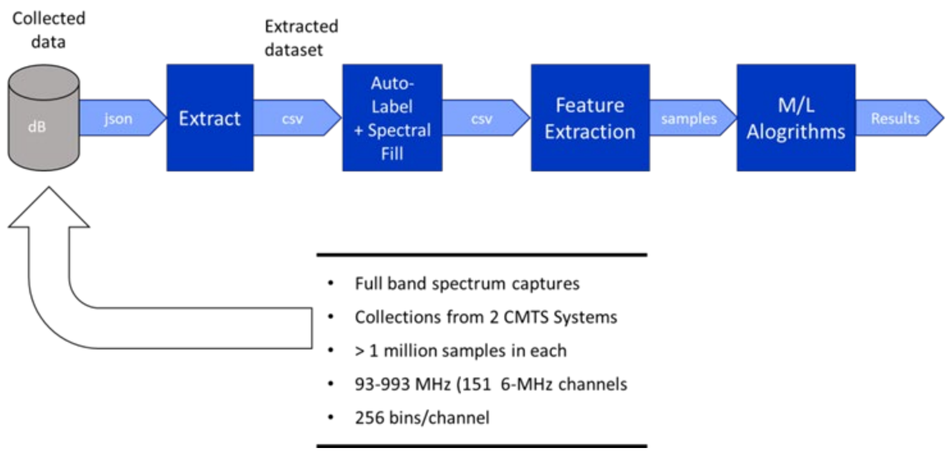

Collected data
dB
json
Extract
Extracted dataset
csv
Auto-Label + Spectral Fill
csv
Feature Extraction
samples
M/L Alogrithms
Results
Full band spectrum captures
Collections from 2 CMTS Systems
> 1 million samples in each
93-993 MHz (151 6-MHz channels
256 bins/channel

Figure 1 - Full Machine Learning Pipeline

5. Data Set Generation

5.1. Extract:

Data queries are generated by an extraction program for data sets of interest. The data is placed in a comma separated value format. Data extraction queries may be based on day and time of samples, device MAC-address, service group or any combination. In this study, all data samples are pulled randomly from available cable modems among all service groups and all available dates and collection times for each CMTS system. Extracted data for each CMTS are held in separate files to evaluate differences in results pertaining to different unique system characteristics of each system.

5.2. Auto-Label + Spectral Fill:

For the purposes of evaluation of ML algorithms we have introduced an auto labeling component. This auto-labeling component uses specified RF metric evaluation to determine the relative quality of an individual spectral sample. For this exercise, the auto-labeler labeled each full-band spectral sample as either good or not good ('impaired'). The full-band capture was evaluated using the following RF-based metrics:

a) Power in one or more channels did not fall within the DOCSIS specification range for received power (-15 dBmv to +15 dBmv)
b) The power level difference from a channel to either adjacent channel was greater than 3.5 dB.
c) A max3 calculation defined as the change in power of any points within a consecutive 3 MHz channel spectrum was greater than 3.5 dB.
d) A max6 calculation defined as the change in power of any points within a consecutive 6 MHz channel spectrum was greater than 3.1 dB.

5

Samples that failed one or more of the above tests were flagged as 'impaired' for the purposes of a machine-learning training set. Spectrum samples that passed all tests were labeled as 'good'. It is important to note that samples labeled 'impaired' for this purpose do not necessarily mean customer services are impaired. A better description might be that the margin between current operating condition and eventual service degradation is less than ideal.

The FBS captures provide data for all channels between 93 MHz and 993 MHz inclusive. Not all channels in this wide range include active video or data services and thus have no transmitted energy in the 6 MHz channel. These channels are labeled as 'unused'. Early tests indicated that unused channels could possibly impact the overall performance of the M/L algorithms in categorizing impairments.

Unused spectrum may also contain ingress from unwanted sources or include other noise sources from the cable plant.

To assess the impact of data associated with unused channels, the auto-labeler incorporates a channel sensing algorithm to generate a list of active channels across the spectrum.   The channel sensing algorithm determines 6MHz channel energy and other attributes to compare with thresholds to determine presence or absence of a channel. The output of the auto-labeler includes a list of frequencies for which active channels are present. The complement of this list is therefore the list of unused 6 MHz channels in this spectrum. This complement list is used to evaluate different techniques to aid the final spectral classification. The options evaluated are:

a) do nothing with the channel gaps and process 'as is' or
b) fill the channel slots that have no transmitted energy with a fixed value of -60 dBmV. A fixed -60dB fill was selected with the notion that the characteristics of a simple 'line' fill may ease the identification burden on the algorithm from unused spectral capture of a real received rf signal which includes random noise power.

Figures 2 and 3 show a typical RF FBS capture and with a -60 dBmV fill respectively.

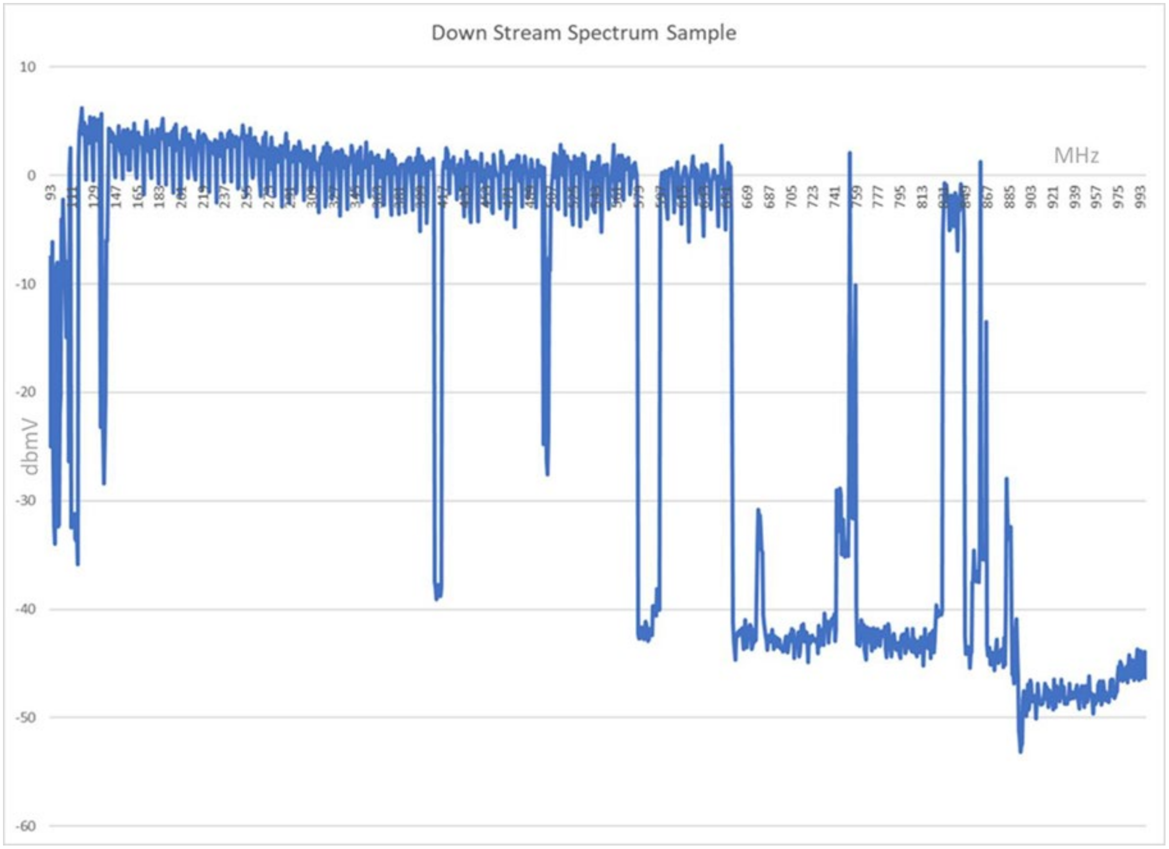


Figure 2 - Example spectrum capture

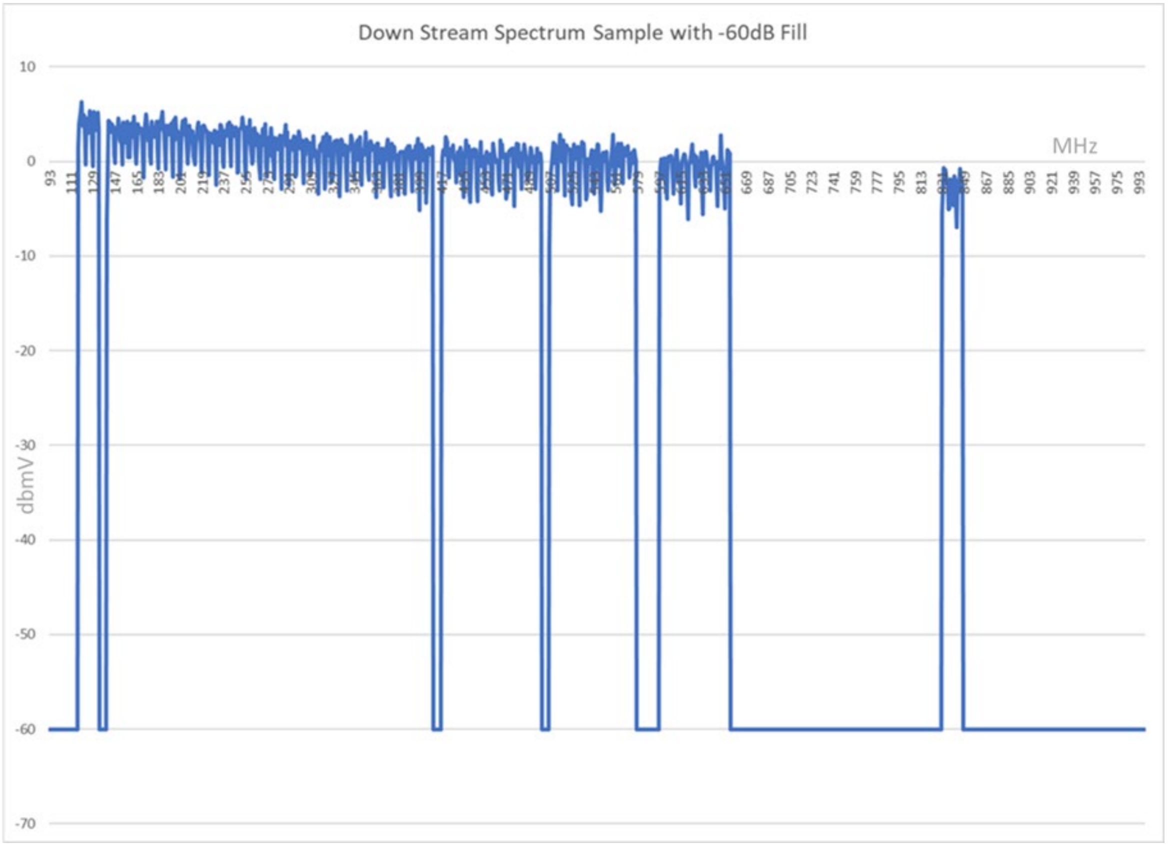


Figure 3 - Example of Spectrum with -60 dB fill

The datasets are taken from field data on operational systems. The number of samples that are labeled 'impaired' are a small subset of the overall number of samples included in the set. Once the labeling is completed, the data sets are trimmed by removing some of the samples labeled 'good' to improve the balance between the number of 'good' labeled samples and 'impaired' labeled samples. The final sets have a slight bias toward 'good' samples.

5.3. Feature Extraction:

Available data sets are fed into a feature extraction program. The objective of the feature extraction algorithms is to reduce the total number of features associated with a spectral sample to something smaller and more explanatory to the output labels. In absence of the feature extraction algorithm the entire dataset is effectively a set of 38k individual features, each spectral bin treated as a data feature. Using a feature extraction algorithm, the total number of features per sample may be reduced by orders of magnitude to allow for faster and lower cost processing of the data.   The reduced size feature sets are used in 3 of the 4 ML algorithms while one of the algorithms uses the FBS spectrum directly.

TSFresh[1] 1is a generic feature extraction library utilized for this purpose based on [4]. No specific domain knowledge is leveraged for feature extraction. TSFresh is an opensource python package for feature extraction of time-series based data but applicable to any uniformly sampled data, in this case

[1] https://tsfresh.readthedocs.io/en/latest/index.html

spectral samples. TSFresh includes a basic set of 788 single value features extracted from each of the data samples where data sample refers to an individual FSB capture.

Some examples of the features calculated by TSFresh include Absolute Energy, all values in the sample are squared and summed to a single value. A second example feature is Max value where only the maximum value of the entire sample is retained. Other features include autocorrelation where the autocorrelations of each sample are taken with all possible lags (for example delay by 1, 2, 3,.. up to n-1 bins) and the results are summed together. Upon the calculation of the TSFresh features, the initial spectral samples are no longer required, the original labels are retained with the newly computed TSFresh feature set.

Principal component analysis (PCA) is another technique that is further applied to the extracted features to reduce the feature set to a smaller set. PCA is a linear algebra technique that accomplishes dimensional reduction using linear transformations of features to determine the optimal linear combinations of features and their contribution to the overall explanation of the sample output labels, discarding features that have insignificant value. An in-depth overview of PCA is available in [5]. In this study, we evaluated the machine learning algorithms using the extracted feature set from TSFresh using PCA to further reduce the feature set. We evaluated data sets with feature extraction using a PCA threshold of 99% for this paper.

6. ML Algorithms

This analysis includes evaluation of the following machine-learning algorithms:

Adaboost – Adaboost is a decision tree based algorithm that makes use of many simple decision tree classifiers. Adaboost utilizes a base classifier, in this case a decision tree classifier. The Adaboost algorithm builds a final classifier using weighted sample data over a series of training passes. The best decision trees in each pass are added to a final weighted decision tree classifier set.

Logistic Regression – In logistic regression, a logit function is used to create a best-fit decision boundary between samples labeled as good or bad resulting in a linear decision boundary. The LR algorithm is optimized using the training data to minimize the error of the fitted data. Samples are compared against the constructed decision boundary and labeled accordingly.

Multi-layer Perceptron – A MLP is the simplest form of neural network that may be used for classification problems. A MLP produces a non-linear decision boundary using a network of simple nodes.

ResNet – ResNet is a convolutional neural network architecture developed for deep-learning [6]. The ResNet architecture solves issues associated with vanishing gradients and accuracy degradation with deep learning architectures. ResNet is the only algorithm that uses FBS samples directly without feature extraction and PCA reduction.

7. Learning Curves

Learning curves can be useful to provide guidance on whether an algorithm is being over trained (i.e., over-fitting) or is not being trained with enough data. To perform a learning curve analysis, a large data set was extracted using System B data. The algorithms were trained using varying amounts of training data from 2k samples to 12k samples. For each level of training, test and training accuracy are recorded. Learning curves for MLP, LR, and AdaBoost algorithms are shown in Figure 4, Figure 5, and Figure 6 respectively. These learning curves indicate that training samples above 10-12k provide little additional gain in the overall testing performance of the algorithms. The learning curves are useful to get a general idea on the training necessary for the algorithms. Note that the learning curves may be different for each of the different data sets – i.e., System A, System B with and without -60db fill. Learning curves are generated using a particular set of hyper-parameters for each of the algorithms. Hyper-parameters are parameters that impact the algorithmic model configuration. Examples of hyper-parameters are the specific method for gradient search, a regularization parameter value, or the specific activation function of a neural network. Because we are doing hyper-parameter tuning for each test the actual hyper-parameters may change from those used in the learning curves. Nevertheless, the learning curves can provide some useful guidance on the data set size, but further testing may be warranted.

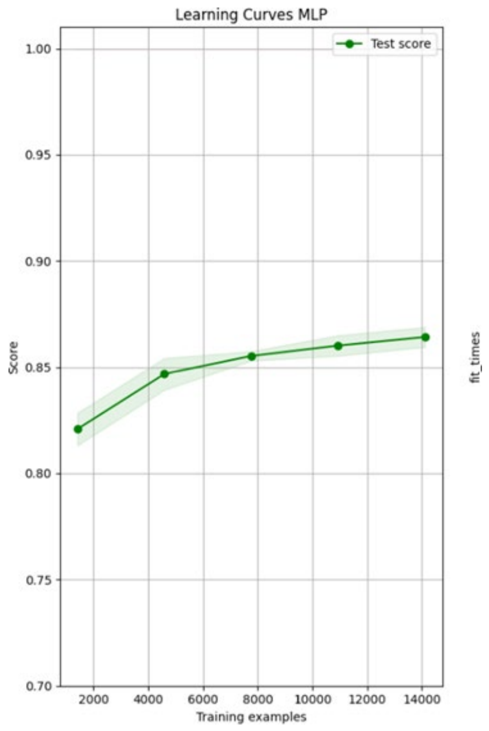


Figure 4 - MLP Learning Curve

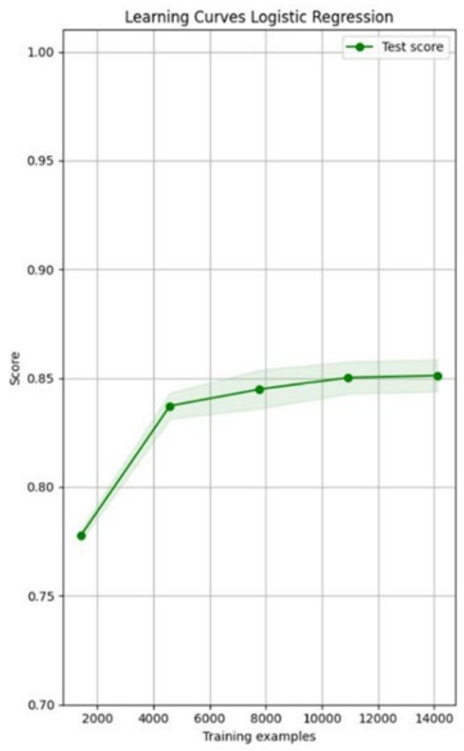


Figure 5 - LR Learning Curve

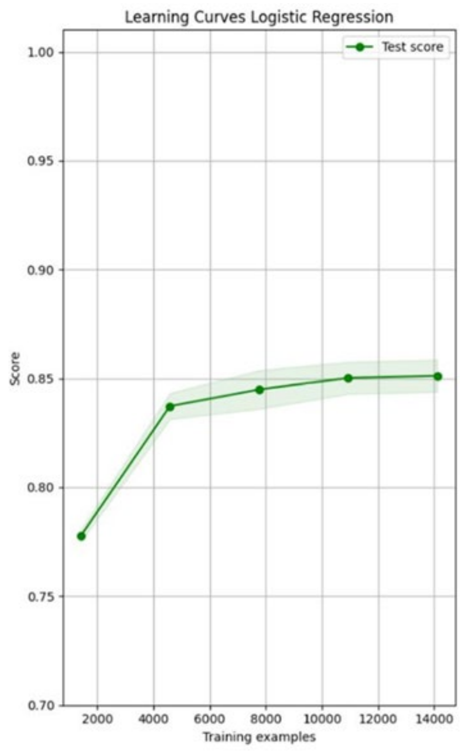


Figure 6 - AdaBoost Learning Curve

8. Data Splitting for Train and Test

For each algorithm, the data set is split into a training set and test set. The split between training and test sets are 75% and 25% respectively of the overall data set.

The training set and test set each consist of the features extracted along with the auto-generated labels. The training set is used to train the machine-learning algorithms while the test set is used once the algorithms are appropriately trained to evaluate the algorithm on similar but different data that it has been trained on. The training set is normalized prior to training to prevent any features from dominating training due to scale.

For each algorithm, hyper-parameters are tuned using both the training set and test sets prior to a final training and test evaluation. Once the test set is completed the algorithm performance is calculated based on the predicted values of the test set results and the actual values as labeled by the auto-label process.

Accuracy and confusion matrix results are recorded for each data set. Description of confusion matrix and associated definitions are in the Appendix, 13.

9. Data Sets

The data sets for this analysis have been extracted from 2 different networks in different geographical locations. Both data sets were taken randomly from each network. The characteristics of the data sets are provided in Table 1 below.

Table 1 - Dataset Characteristics

System	Samples	Unused Spectrum (%)	Bias (%)
A	11226	24.65	51.283
B	16296	32.01	51.283

Table 1 lists the number of samples, initial bias, and amount of unused spectrum based on the aggregate of all the samples in that set. The Unused Spectrum is an aggregate value calculated by counting the number of bins in each spectrum capture that have been filled during the Spectral Fill process divided by the total bins in each full spectrum capture, each individual sample may have more or less unused spectrum than the overall aggregate percent. The Bias is calculated by summing the number of labels = '1' ('impaired') divided by the total number of samples in the set and subtracting from 1. A number greater than 50% reflects the situation that the Bias represents a slightly greater number of 'good' samples than 'impaired' samples in each set.

Each of the algorithms were run across data sets from two different systems to evaluate performance among the algorithms and differences among the data sets. Feature sets were created using Principal Component Analysis (PCA) with only features meeting a prescribed threshold of 99% used. For reference, the full TSFresh feature set consists of 788 different feature metrics.

For each algorithm, the ML classifier was trained using the training set. The specific approach to training differs depending on the ML algorithm. For example, with MLP, the entire training set is passed through the algorithm, called an epoch, after which a cost function is used to update the MLP weights using a statistical gradient descent search and learning rate. After many epochs, the MLP weights approach the optimal values for performance. Once training is complete, the test set is passed through the MLP. The specifics of training for each of the algorithms are different, the general concept of optimizing the performance using the training set and evaluating using the test set is the same. For each sample in the test set, an output prediction ('good' or 'impaired') is generated. The prediction is then compared with the actual label for that sample. Accuracy is defined as the total number of correct output predictions divided by the total number of samples in the test set.

10. Results

Figure 7 and Figure 8 show the results using the 99% PCA feature set for systems A and B in graphical format respectively while Table 2 and Table 3 provide the results in tabular format. Table 2 and Table 3 include an additional column providing the difference in accuracy between the case of no fill and that of -60dB fill. The number is referenced to the case of no-fill so that a positive number represents the case where a -60dB fill improved the results. Table 4 shows the number of PCA extracted features for each dataset.

In reviewing the results, it is apparent that all ML algorithms evaluated performed significantly better than simply categorizing all samples as 'good' which would provide about a 51% accuracy.

Second, the performance of System A and System B were not identical, System A generally tended to have better results than System B across all the ML algorithms. This observation could be related to the fact System A had fewer unused channels than System B.

At the individual algorithm level, ResNet tended to have better results than the other algorithms with MLP being a close second. These algorithms are based on neural network models capable of creating complex decision boundaries.

With the exception of LR, all the algorithms benefitted from the synthetic -60db fill with Adaboost performance bumping by over 4% when using System B data. One possible explanation for LR is that LR generates a linear decision boundary. Given the feature set extracted, the actual decision boundary is likely not linear which may limit the general overall performance of LR. Additionally, the level of improvements from using the synthetic -60dB fill were greater in System B. This may also be due to the fact that System B included a greater amount of unused spectrum than System A making the synthetic fill more impactful across the samples in System B.

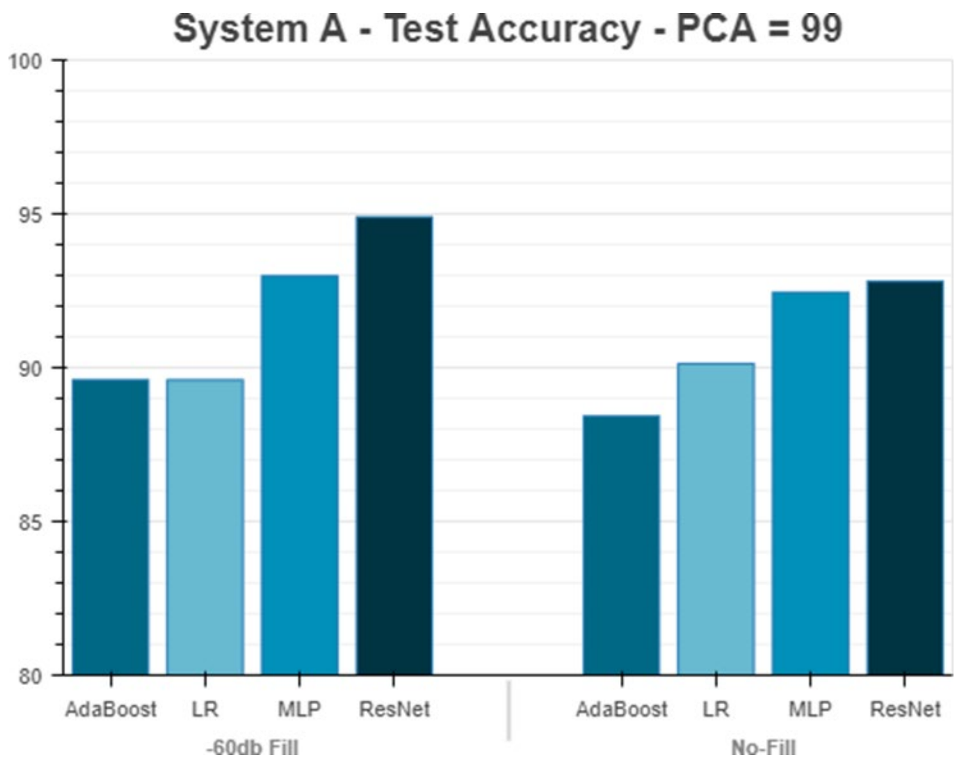


Figure 7 - System A Results

Table 2 - System A Accuracy

SYSTEM A

	NoFill	-60db Fill	Diff
ADABOOST	88.42	89.6	1.18
LR	90.13	89.6	-0.53
MLP	92.45	92.98	0.53
RESNET	92.8	94.9	2.1

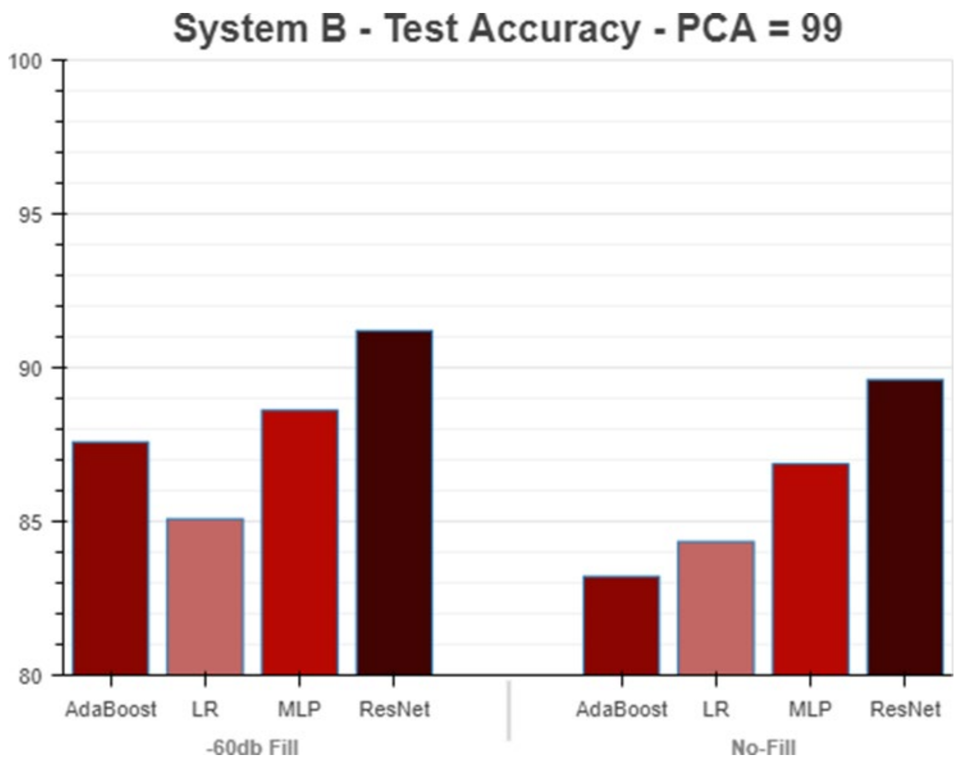
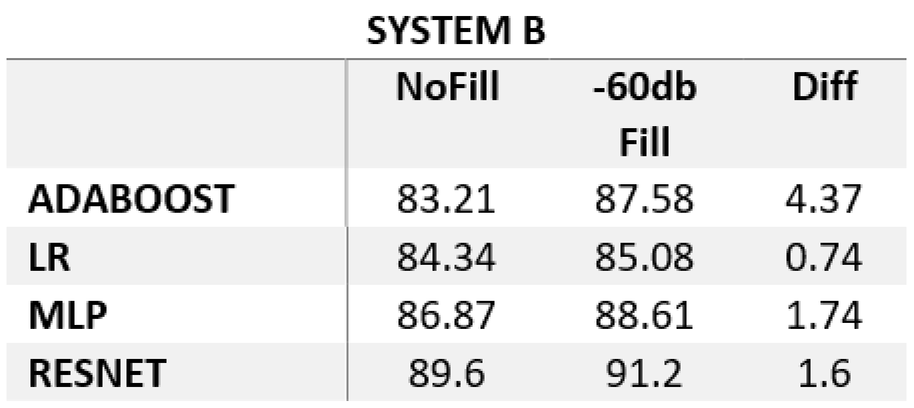


Figure 8 - System B Results

Table 3 - System B Accuracy



SYSTEM B

	NoFill	-60db Fill	Diff
ADABOOST	83.21	87.58	4.37
LR	84.34	85.08	0.74
MLP	86.87	88.61	1.74
RESNET	89.6	91.2	1.6

Table 4 - Number of features by data set, PCA = 99

System	Fill	Extracted Features
A	None	251
B	None	258
A	60db	172
B	60db	152

11. Summary

This paper evaluated the use of several M/L algorithms for the case of categorization of full band spectral data from DOCSIS systems to categorize 'impaired' spectrum from spectrum which is not impaired. The M/L algorithms evaluated are supervisory based algorithms where labeling for training and testing was provided using traditional RF signal processing approaches. Data sets were generated from field FBS captures from 2 different CMTS with different configurations and network characteristics. The results of applying the M/L algorithms to the data show that the use of M/L algorithms provided significant gain in identifying impaired spectrum from non-impaired spectrum according to the labeling system used. In addition, using a spectral fill algorithm consisting of a simple constant value for unused spectrum improved the categorization accuracy by over 4% with the Adaboost algorithm and nearly 2% with the ResNet algorithm in one case. In general, neural network based algorithms tended to provide the best overall prediction accuracy. The system with the least unused spectrum exhibited the best performance while using the synthetic -60dB fill provided a greater performance improvement for the system with the greater amount of unused spectrum.

12. Implementing A Closed Loop Approach to Managing Spectrum

This paper has analyzed the potential for using full band spectrum capture data for the identification of potential impairments in a DOCSIS plant. As with any machine learning exercise, many steps of data processing and cleansing are needed prior to actual training and testing of the algorithms. Moreover, feature extraction and training of models are both time and resource consuming processes. Fortunately, once a model is trained, using the model to classify actual samples is a much quicker process.

Figure 9 depicts the various processes necessary in a closed loop system. Given a set of CMTS' managing a DOCSIS network, a Data Extractor must periodically extract the full-band spectrum data from the devices in the network. This can be a periodic task performing a PNM FSB test on each device once per day, preferably during early morning low utilization times, storing the results in a database. The classification and clustering system extracts data on a daily process to classify each device to be potentially problematic or good. Additional clustering may also be employed using meta data from the captures such as service group identifiers. This would allow grouping devices with common network topology to help localize issues within the network distribution system. The classification system generates a report indicating devices that may be problematic and any indications of potential network trouble spots. Finally, Model Training may be performed periodically using updated data collected and placed in the database. Updating models may be necessary in the event of network changes. As new training takes place, model evaluation should be done from the prior model to the new model to obtain some indication on the dynamics governing the value of re-training the network.

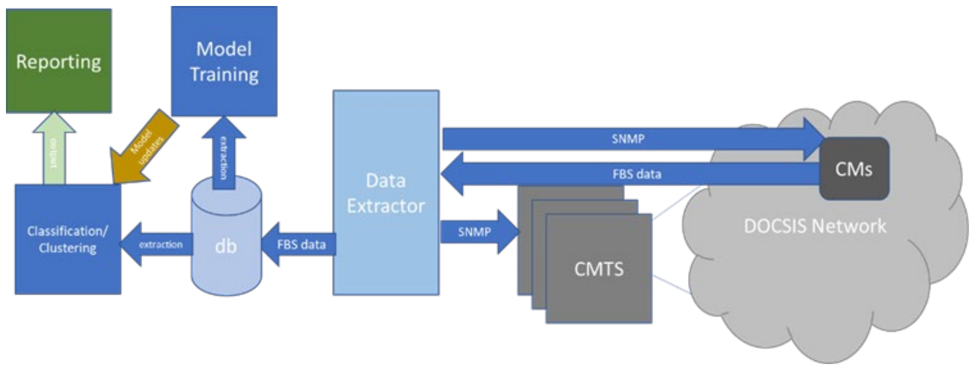
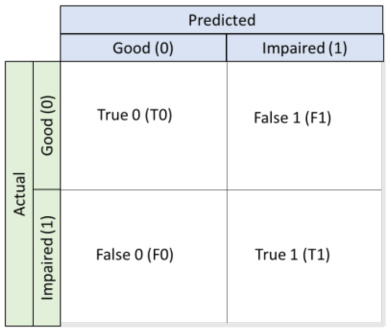


Figure 9 - Closed loop implementation using ML

13. Appendix

Note on Accuracy and Confusion Matrix Definitions A typical confusion matrix is show in Figure 10. .



		Predicted	
		Good (0)	Impaired (1)
Actual	Good (0)	True 0 (T0)	False 1 (F1)
	Impaired (1)	False 0 (F0)	True 1 (T1)

Figure 10 - Confusion Matrix

In this paper, samples for spectrum are labeled 0 for 'good' spectrum and labeled 1 for 'impaired spectrum. The confusion matrix is given in this example as Good samples (0) or Impaired samples (1).

The definitions used in this paper are as follows:

Accuracy = (T0 + T1) / (T0 + F0 + T1 + F1). Accuracy is the total number of correct classifications over the entire set of tested samples.

Precision = T1 / (T1 + F1). Precision is useful to describe the relevancy of the 'impaired' samples resulting from the testing predictions. For example, if all samples classified as 'impaired' by the algorithm are actually labeled 'impaired' the Precision value would be 100%. If 25% of those classified as 'impaired' by the algorithm were 'good' samples mistakenly classified as 'impaired', the accuracy would be 75%.

Recall = T1 / (T1 + F0). Recall is useful to describe the completeness of the classified set as determined from the algorithm. If the results classified as 'impaired' included every 'impaired' sample in the actual data set, the Recall would be 100%. If 25% of the actual 'impaired' samples were classified as 'good' they would not appear in the 'impaired' list. The resulting Recall in this case would be 75%.

Note that 100% Precision does not imply 100% Recall or vice-versa.

14. Accuracy and Confusion Matrix data

The following tables provide the confusion matrix details for each of the trials. In the following tables, the Precision and Recall numbers are provided in reference to 'impaired' classifications. The of the table values are prefixed by the algorithm, e.g., LRT0 is the True 0 value for the Logistic Regression, ABF0 is the False 0 for the AdaBoost algorithm, and MLPT1 is the True 1 value for the MLP algorithm.

1. Confusion Matrix Table for Logistic Regression

System	Interp	PCA	Samples	LRT0	LRF1	LRF0	LRT1	LRPrec	LRRecall
A	None	99	11226	1308	116	161	1222	91.3%	88.4%
B	None	99	16296	1734	343	295	1702	83.2%	85.2%
A	60db	99	11226	1287	137	155	1228	90.0%	88.8%
B	60db	99	16296	1766	310	298	1700	84.6%	85.1%

2. Confusion Matrix Table for AdaBoost

System	Interp	PCA	Samples	ABT0	ABF1	ABF0	ABT1	ABPrec	ABRecall
A	None	99	11226	1274	150	175	1208	89.0%	87.3%
B	None	99	16296	1691	386	298	1699	81.5%	85.1%
A	60db	99	11226	1297	127	165	1218	90.6%	88.1%
B	60db	99	16296	1814	262	244	1754	87.0%	87.8%

3. Confusion Matrix Table for MLP

System	Interp	PCA	Samples	MLPT0	MLPF1	MLPF0	MLPT1	MLPPrec	MLPRecall
A	None	99	11226	1326	98	114	1269	92.8%	91.8%
B	None	99	16296	1802	275	260	1737	86.3%	87.0%
A	60db	99	11226	1330	94	103	1280	93.2%	92.6%
B	60db	99	16296	1841	235	229	1769	88.3%	88.5%

Abbreviations

DB	Database
CM	Cable modem
CMTS	Cable modem termination system
CNN	Convolutional neural network
CPE	Consumer premise equipment
dBmv	Decibels relative to 1 millivolt
DOCSIS	Data over cable system interface specification
DSSA	Down stream spectrum analysis
FBS	Full band spectrum
FTP	File transfer protocol
LR	Linear regression
MHz	Megahertz
ML	Machine learning
MLP	Multi-layer Perceptron
OFDM	Orthogonal Frequency Division Multiplexing
PCA	Principle Component Analysis
PNM	Proactive Network Maintenance
QAM	Quadrature Amplitude Modulation
RF	Radio frequency
SCQAM	Single channel QAM
SNMP	Simple Network Management Protocol

Bibliography & References

[1] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar and P.-A. Muller, "Deep Learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, no. 4, pp. 917-963, 2019.
[2] K. Sundaresan and J. Zhu, "Access Network Data Analytics (Machine Learning Applied to Cable Access Data)," 2017.
[3] J. Ferreira, H. Maher, K. Subramanya, B. Santangelo and D. Rice, "Convolutional Neural Networksfor Proactive Network Management - Developing Machine Learning Models to Detect and Classify Impairments in D3.1 OFDM Channels," 2020.
[4] M. Christ, A. W. Kempa-Liehr and M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications," 2017.
[5] J. Leskovec, A. Rajaraman and J. D. Ullman, "Dimensionality Reduction," in Mining of Massive Datasets, 2014, p. 405.
[6] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," Microsoft Research, 2015.