Dynamic Drift-Adaptive Ensemble-Based Quality of Transmission Classification Framework in OTN

Huy Quang Tran[1,3], Javier Errea[1,2], Van-Quan Pham[1], Dominique Verchere[1], Adlen Ksentini[2], and Djamal Zeghlache[3], *Member, IEEE*

[1]NOKIA Bell Labs, France, US
[2]EURECOM Research Institute, Biot, France
[3]Telecom SudParis, Institut Polytechnique de Paris, France
E-mail: huy.h.tran@nokia-bell-labs.com

ABSTRACT

We introduce Dynamic Drift-Adaptive Ensemble-based Quality of Transmission (QoT) Classification Framework (DAEQoT) to effectively verify the feasibility of a lightpath while maintaining high prediction accuracy in a dynamic Optical Transport Network (OTN) scenario. This framework adopts the Early Drift Detection Method (EDDM) to identify any significant increase in the prediction error. Moreover, Ensemble Learning is implemented to enhance the accuracy of the Master QoT model by combining other Supportive QoT classifiers when an early drift warning is reported. Thus, prediction error and complete model retraining are mitigated. DAEQoT achieves the best accuracy of 98.56% and reduces the execution time up to 52.81% compared to state-of-the-art offline and online machine learning approaches.

Keywords: QoT classification, drift adaptation, online ensemble learning.

1. INTRODUCTION

With the rise of 5G and upcoming 6G services characterized by virtualized and disaggregated multi-vendor environments operating in the cloud, network infrastructure has become more diverse and distributed. As a result, E2E network orchestration and assurance must now consider a broader range of network components, technologies, domains, and their interactions, i.e., Radio Access, Transport, and Core. In the transport domain, VPN technology is usually used in mobile transport networks for years; underpinning VPN and packet switching technology is the DWDM optical transmission technology. Optical networking plays a role in 5G/6G mobile transport and can also participate in network slicing. One example is assigning separate optical wavelengths to individual network slices. To enable E2E network services, the primary requirements are abilities to translate the intent of a slice consistently and automatically, like mission-critical or low-latency services, and map that intent to a set of transport performance requirements and resource allocations at the VPN and packet switching layers, as well as in the optical domain. Therefore, an E2E slice intent can be decomposed into a set of domain slice intents, including the optical network intent (e.g., lightpath or wavelength). On the other hand, Machine Learning (ML) shows many benefits in Intent-Based Networking for Optical Networks [1]. Several ML models (e.g., QoT classification/prediction, Anomaly Detection...) can be associated with an optical network intent to support the automated management operation of that optical intent (lightpath), However, there are also many challenges in managing those ML models. One of the critical challenges is to monitor and maintain the performance (e.g., prediction accuracy...) of ML models under data distribution shift or model objective change as the network performance changes over time.

Quality of Transmission (QoT) classification using ML models has become a hot topic since it shows more effective and accurate than numerical and analytical approaches (i.e., the split-step Fourier method, the Gaussian Noise model) [2]. We consider the requirement to verify the feasibility of a new lightpath before the provisioning phase. The QoT violation assurance imposes extra complexity because there might be many active connections that need QoT value verification. Most QoT ML models are implemented with the assumption that the training and testing dataset follow the same pattern. However, it is no longer the case in the dynamic Optical Transport Network (OTN) scenario where data distribution in the target environment may differ from the learning one. Thus, QoT ML models could experience a performance degradation or drift. In addition, the lack of drift adaptation mechanism prevents the model to dynamically improve its performance, which causes a complete update on new samples. Model retraining requires an advanced scheme to organize new online data, tune hyperparameters, record and serve various versions of the model.

We introduce Dynamic Drift-Adaptive Ensemble-based Quality of Transmission Classification Framework (DAEQoT) to overcome this problem. DAEQoT allows the use of a state-of-the-art batch-learning ML model, while combining other online methods to form an ensemble, which enhances the prediction accuracy in the event of a potential drift. This approach uses the Early Drift Detection Method to monitor the performance of the models. Evaluations show significant improvement in prediction accuracy, as compared to both online and offline techniques. Moreover, it is able to avoid severe drift events without the need for a complete retraining of the model.

2. BACKGROUND

2.1 Model Drift

Model drift occurs when the performance of a ML model degrades over time. There are various causes such as data distribution shift, model objective change, etc. There are two main categories: data drift and concept drift. **Data drift** occurs when the distribution of the features in prediction is different from the training phase. For example, the traffic matrix of connectivity requests could be shifted from a high number of low-capacity demands (i.e., 100 Gbps, 200 Gbps) to more high-capacity ones (i.e., 300 Gbps, 400 Gbps) due to the bandwidth requirement of 5G and Beyond 5G services. **Concept drift** indicates that there is a change in the underlying relationships between features and outcomes. Particularly, a defective optical receiver would report a failure although the receiving Optical Signal-to-Noise ratio (OSNR) is well above the threshold. Hence, label are reported incorrectly which leads to the degradation of the ML model's performance.

2.2 Drift Detection

There are two main methods to detect drift: data monitoring (or statistical method) and performance monitoring (or learner-based method**). Statistical or hypothesis tests** are used to detect actual changes in data distribution. There are popular methods such as Population Stability Index (PSI), Kullback-Leibler (KL) divergence, Jensen-Shannon (JS), Kolmogorov-Smirnov (KS) test, etc. Particularly, PSI measures how much a population has shifted over time. In addition, KS test quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples. **Monitoring model performance metrics** is the most fundamental approach for drift detection. There are common metrics to monitor such as confusion matrix, accuracy, recall, F1 score, etc. Some approaches focus on the error rate and use the error rate-based drift detection method. For example, the Drift Detection Method (DDM) algorithm can be used to detect any significant increase in error rates. We adopt the Early Drift Detection Method (EDDM) [3] as the performance-based drift detection algorithm that determines the occurrence of model drift by monitoring the degree of model performance degradation to improve the detection in presence of gradual drift; while, keeping a good performance with abrupt drift. EDDM examines the average distance between two prediction errors ($\bar{p}_t$) and its standard deviation ($\bar{s}_t$) instead of the number of errors. We obtain $\bar{p}_{max}$ and $\bar{s}_{max}$ when ($\bar{p}_t + 2.\bar{s}_t$) reaches its maximum. The algorithm considers two parameters α and β as the warning and drift threshold, respectively. Particularly, EDDM will signal Warning if $\frac{\bar{p}_t+2.\bar{s}_t}{\bar{p}_{max}+2.\bar{s}_{max}} < \alpha$; or, Drift if $\frac{\bar{p}_t+2.\bar{s}_t}{\bar{p}_{max}+2.\bar{s}_{max}} < \beta$.

2.3 Drift Adaptation

The drift adaptation procedure is referred to as automated model updates in the network data analytics automation process, as its main purpose is to improve model performance by updating the learning model. Drift adaptation aims at improving model performance by automatically updating the model in the data analytics automation process. After identifying a concept drift, learning models should be able to adapt to new concepts and enhance model performance. Existing drift-adaptive learning techniques fall into two primary categories: incremental learning and ensemble learning techniques.

Incremental learning is the process of learning each incoming data sample in chronological order and partially updating the learner. Hoeffding Tree [4] is a basic incremental learning method that employs the Hoeffding inequality to determine the minimum number of data samples necessary for each split node, thus updating nodes to adapt to new samples. **Ensemble online learning models** are advanced drift- adaptive learning methods that integrate the output of multiple base learners for performance improvement.

3. DRIFT-ADAPTIVE ENSEMBLE-BASED QUALITY OF TRANSMISSION CLASSIFICATION

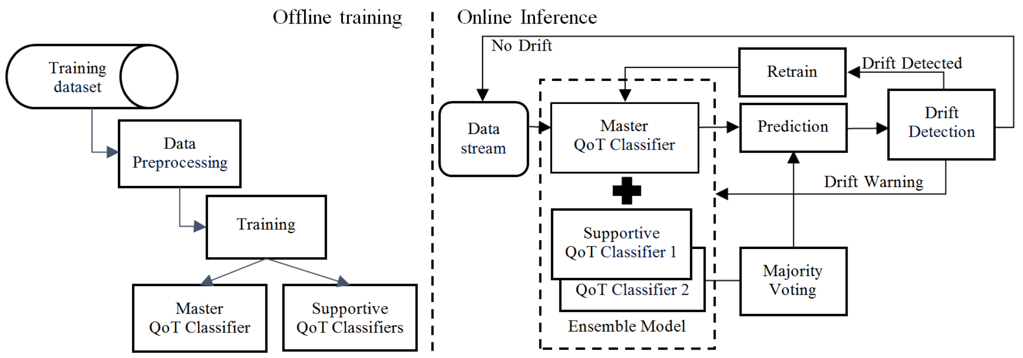

Figure 1. Dynamic drift-adaptive ensemble-based quality of transmission classification framework.

The framework has two main components: a set of QoT Classifiers and a drift detection mechanism. There are two phases involved in this framework: offline training and online inference. In the training phase, both Offline and Online ML models are considered and trained with the same learning set. We choose CatBoost [5], which is a ML framework based on gradient-boosted decision trees, as the Master QoT Classifier. In addition, online ML models such as Adaptive Random Forest (ARF), Aggregated Mondrian Forest (AMF), Stream Random Patches (SRP) and Leverage Bagging (LB) are considered as Supportive QoT Classifiers for the Master. While CatBoost is an offline model, which can only be trained in batch, those online models allow to perform incremental learning along the data stream. We select EDDM presented in Section 2.2 as the drift detection mechanism in the online phase since it can provide no drift, warning and drift detected information. The architecture is presented in Figure 1.

Moreover, Algorithm 1 describes the operation of DAEQoT in the online phase. The system begins at "No drift" phase ($Signal = 0$) and $\boldsymbol{EDDM}$ is initiated with the warning α and drift threshold β. $\boldsymbol{DAEQoT}$ uses only the Master QoT Classifier $\boldsymbol{M}$ to predict the QoT $\tilde{\boldsymbol{y}}_i^{\boldsymbol{M}}$ of each sample $\boldsymbol{s_i}$ from the data stream, based on the feature vector $\boldsymbol{x_i}$. Next, the true label $\boldsymbol{y_i}$ and the predicted one $\tilde{\boldsymbol{y}}_i^{\boldsymbol{M}}$ are fed to $\boldsymbol{EDDM}$ to compute the average distance between two prediction errors and its standard deviation. Supposed that the prediction accuracy of $\boldsymbol{M}$ degrades after a while, the system turns into the "Warning" phase signaled by $\boldsymbol{EDDM}$. Since the performance has decreased, $\boldsymbol{DAEQoT}$ needs to combine $\boldsymbol{M}$ with other Supportive QoT Classifiers $\boldsymbol{P}$. The ensemble predicted label $\tilde{\boldsymbol{y}}_i^{\boldsymbol{DAEQoT}}$ is derived from the prediction of each classifier in $\boldsymbol{DAEQoT}$ by conducting the Majority Voting mechanism. This output and the true label is then fed to $\boldsymbol{EDDM}$ following the same process. In addition, all samples in the Warning phase are recorded in batch $\boldsymbol{B}$ as new learning data to retrain $\boldsymbol{M}$ at a later phase. On the other hand, since classifiers in the Supportive set $\boldsymbol{P}$ are online learners, they can fit their models in an incremental manner with each sample. If the performance of $\boldsymbol{DAEQoT}$ continues to decay, a complete drift may occur. In this phase, $\boldsymbol{M}$ is retrained using the recorded samples from $\boldsymbol{B}$. The system is reset to the initial "No drift" phase.

```
Algorithm 1: Drift-adaptive ensemble-based QoT classification
     **Input:**
     $S$: data stream
     $Signal$: the output of EDDM ($Signal \in \{0,1,2\}$)
     $M$: Master QoT Classifier
     $P$: set of k Supportive QoT Classifiers
     $B$: Batch to store samples in Warning zone
     **Output:**
     $DAEQoT$: Drift-Adaptive Ensemble-based QoT Classifier
     **Procedure:**
     /* Initialize */
1    $DAEQoT \leftarrow M$ /* Let $DAEQoT$ contain only $M$ */
2    $EDDM_{\alpha,\beta}$
3    $B \leftarrow \emptyset$
4    $Signal \leftarrow 0$ /* No drift */
5    **foreach** $s_i \in S$ **do**:
6        **if** ($Signal == 0$) **then**
7            $\tilde{y}_i^M \leftarrow M(x_i)$
8            $Signal \leftarrow EDDM(y_i, \tilde{y}_i^M)$
9        **else if** ($Signal == 1$) **then** /* in Warning zone */
10           $DAEQoT \leftarrow M \cup P$
11           $\tilde{y}_i^M \leftarrow M(x_i)$
12           $\{\tilde{y}_i^{P_j}\} \leftarrow \{P_j(x_i)\}$ where $j = \{1,..,k\}$
13           $\tilde{y}_i^{DAEQoT} \leftarrow MajorityVoting(\tilde{y}_i^M, \tilde{y}_i^{P_1},..,\tilde{y}_i^{P_k})$
14           $Signal \leftarrow EDDM(y_i, \tilde{y}_i^{AEQC})$
15           $B \leftarrow B \cup (x_i, y_i)$
16           $\{P_j\} \leftarrow \{P_j(x_i, y_i)\}$ where $j = \{1,..,k\}$
17       **else** /* Drift detected */
18           $M \leftarrow M(B)$ /* Retrain $M$ on collected batch $B$ */
19           $B \leftarrow \emptyset$
20           Reset $EDDM_{\alpha,\beta}$
21           $Signal \leftarrow 0$
22       **end**
23   **return:** $DAEQoT$
```

4. EXPERIMENTS AND EVALUATIONS

The proposed framework was implemented by extending CatBoost [5] and River [6] Python library. We used 20000 samples from the Lightpath Dataset 03 in [7] and split them into the learning (20%) and testing (80%) set. All ML models are trained offline with the same learning set and able to achieve the accuracy of more than 95%. Subsequently, ML models are put in the online inference, where each sample in the testing set is fed sequentially to the models. Since SRP and LB provide the best accuracy during the offline training phase, we choose them as two Supportive models, together with the Master, to form DAEQoT.

Figure 2 and 3 depict the performance of the Master QoT Classifier without Ensemble learning and model retraining (Offline Master) and our DAEQoT approach. DAEQoT outperforms the Master QoT Classifier in accuracy and the capability to avoid severe drift. Without any drift adaptation method, the Offline Master encounters drift three times. On the other hand, DAEQoT never reaches the drift threshold thanks to the Drift-Adaptive Ensemble-based mechanism, which enhances its performance and mitigates drift. Figure 4 shows the performance comparison of the proposed framework with other approaches namely ARF, AMF, SRP, LB, Offline and Online Master. The Online Master follows the same principle as the offline one; however, it is retrained in batch mode at every drift point. According to *Table 1*, DAEQoT clearly shows the best performance (98.56%) among approaches, while having a reasonable execution time. The average accuracy of Leverage Bagging (98.47%) is slightly smaller than DAEQoT. However, it takes nearly double the time for Leverage Bagging (3m51), as compared to DAEQoT (1m49), to reach such accuracy. Both Online and Offline Master achieve a reliable prediction of 97.93% and 97.83%, respectively. Since Online Master is updated with new samples during the online operation, it can adapt to the change of the data stream, which leads to a higher accuracy. Moreover, both methods are based on offline ML model, so they can execute within a short amount of time (under 11s). Other online ML models such as ARF, AMF, SRP are not as good as the offline ones in both accuracy and time measurements.

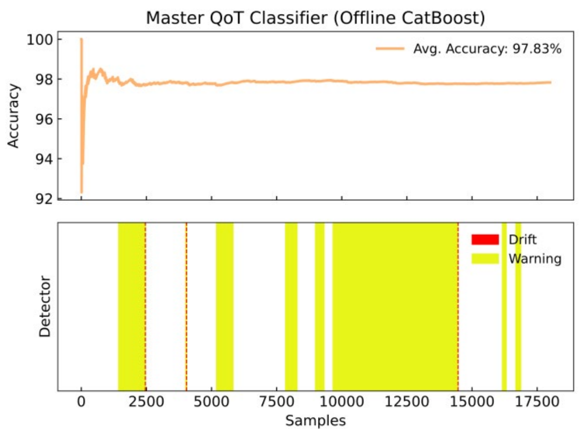

Figure 2. Master QoT classifier.

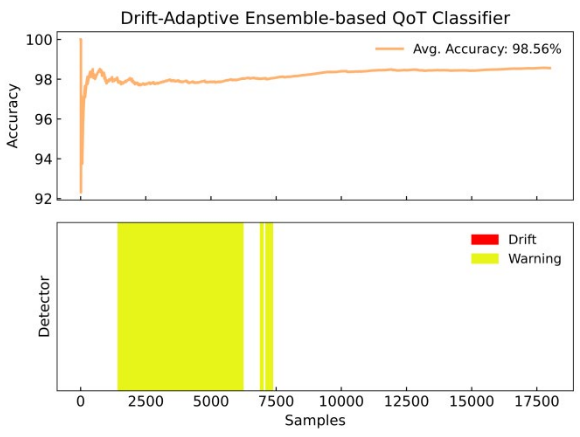

Figure 3. Drift-adaptive ensemble-based QoT classifier.

Table 1. Performance of Drift Adaptation methods.

Model	Accuracy	Execution time
ARF	95.22%	16.7s
AMF	96.26%	14.6s
SRP	96.95%	13.9
Master QoT Classifier (Offline)	97.83%	7.7s
Master QoT Classifier (Online)	97.93%	10.8s
LB	98.47%	3min 51s
DAEQoT	**98.56%**	**1min 49s**

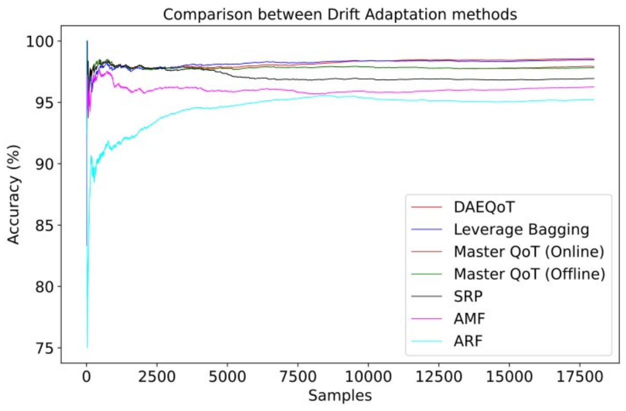

Figure 4. Comparison between drift adaptation methods.

5. CONCLUSIONS

QoT Classification models are trained and used under the assumption that the training and target environment are identical. However, it is no longer the case in future dynamic elastic optical networks. Particularly, a change in request capacity distributions, or undetected defect of an optical receiver, could be the source of model performance degradation. For that reason, the Dynamic Drift-Adaptive Ensemble-Based Quality of Transmission Classification Framework (DAEQoT) is introduced, which is capable of providing a high prediction accuracy and being robust to model drift. This framework adopts the Early Drift Detection Method and Ensemble Learning mechanism to mitigate the prediction error and avoid complete model retraining. DAEQoT achieves the best accuracy of 98.56% among state-of-the-art offline and online machine learning models and reduces the execution time by 52.81% compared to Leverage Bagging online method.

ACKNOWLEDGEMENTS

This work is partly supported by the European Commission via the Horizon Europe/JU SNS project Hexa-X-II (GA no. 101095759).

REFERENCES

[1] L. Velasco, S. Barzegar, F. Tabatabaeimehr, and M. Ruiz, "Intent-based networking and its application to optical networks [Invited Tutorial]," *Journal of Optical Communications and Networking,* vol. 14, pp. A11-A22, 2022.

[2] C. Rottondi, L. Barletta, A. Giusti, and M. Tornatore, "Machine-learning method for quality of transmission prediction of unestablished lightpaths," *Journal of Optical Communications and Networking,* vol. 10, pp. A286-A297, 2018.

[3] M. Baena-García, J. Campo-Ávila, R. Fidalgo-Merino, A. Bifet, R. Gavald, and R. Morales-Bueno, "Early drift detection method," Jan. 2006.

[4] G. Hulten, L. Spencer and P. Domingos, "Mining time-changing data streams," *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Jul. 2001.

[5] A. V. Dorogush, V. Ershov, and A. Gulin, CatBoost: Gradient boosting with categorical features support*,* 2018.

[6] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem *et al.*, "River: Machine learning for streaming data in Python," 2021.

[7] G. Bergk, B. Shariati, P. Safari, and J. K. Fischer, "QoT Dataset Collection," [Online]. Available: https://www.hhi.fraunhofer.de/networkdata.