Research Article | Vol. 27, No. 8 | 15 Apr 2019 | OPTICS EXPRESS 10631
Optics EXPRESS

Dirty-data-based alarm prediction in self-optimizing large-scale optical networks

BOYUAN YAN,[1] YONGLI ZHAO,[1,*] SABIDUR RAHMAN,[2] YAJIE LI,[1] XIAOSONG YU,[1] DONGMEI LIU,[3] YONGQI HE,[4] AND JIE ZHANG[1]

[1]*State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*University of California, Davis, Davis, CA, USA*
[3]*State Grid Information & Telecommunication Company, Beijing 100761, China*
[4]*State Key Laboratory of Advanced Optical Communication Systems Networks, Peking University, Beijing 100871, China*
**yonglizhao@bupt.edu.cn*

Abstract: Machine-learning-based solutions are showing promising results for several critical issues in large-scale optical networks. Alarm (caused by failure, disaster, etc.) prediction is an important use-case, where machine learning can assist in predicting events, ahead of time. Accurate prediction enables network administrators to undertake preventive measures. For such alarm prediction applications, high-quality data sets for training and testing are crucial. However, the collected performance and alarm data from large-scale optical networks are often dirty, i.e., these data are incomplete, inconsistent, and lack certain behaviors or trends. Such data are likely to contain several errors, when collected from old-fashioned optical equipment, in particular. Even after appropriate data preprocessing, feature distribution can be extremely unbalanced, limiting the performance of machine learning algorithms. This paper demonstrates a Dirty-data-based Alarm Prediction (DAP) method for Self-Optimizing Optical Networks (SOONs). Experimental results on a commercial large-scale field topology with 274 nodes and 487 links demonstrate that the proposed DAP method can achieve high accuracy for different types of alarms.

1. Introduction

With the rapid increase in data-traffic load, new technologies for creating higher capacity optical networks are on the rise [1,2]. This leads to higher economic loss if a single fiber link is down, even for few seconds. Therefore, accurate alarm prediction is becoming increasingly critical for network operators. In recent years, Machine Learning (ML) has been used in various network use-cases [3,4] because of its high accuracy in predicting future events. Alarm prediction has been explored in literature. Ref [5]. proposed an alarm prediction method using support vector machine and double exponential smoothing. Ref [6]. proposed a ML technique for predicting whether the bit-error-rate of an unestablished lightpath is within the normal range, based on the traffic, routing, and modulation format. Ref [7]. proposed a Gaussian process classifier to predict the probability of alarm for each optical link. All these ML methods realize high prediction accuracy under the assumption of availability of well-handled training and testing data sets.

However, network performance and alarm data collection is a difficult task for telecom operators because vendor-equipment differ with respect to the device type, device version, network management system, etc. For old-fashioned-malfunctioning equipment, long-term data collection has significant limitations in terms of the memory, bandwidth, measurement accuracy, etc. In addition, the collected data are generally dirty. Dirty data, also known as rogue data, include missing data, wrong data, and nonstandard representations of the same

#355634 https://doi.org/10.1364/OE.27.010631
Journal © 2019 Received 18 Dec 2018; revised 15 Mar 2019; accepted 24 Mar 2019; published 2 Apr 2019

data [8]. Our study considers this special case of dirty data for alarm prediction in optical networks.

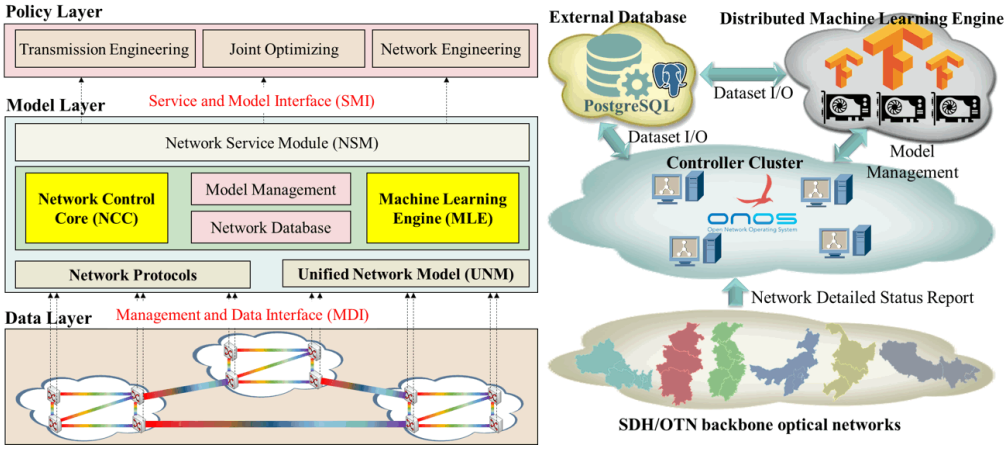
Network alarms can be classified into several types: fiber cut, low output power, lightpath failure, etc. In practice, the ratio of network alarm records caused by different reasons is highly imbalanced which may be unsuitable for ML training. Because this imbalance may lead to unsatisfying results by over-focusing on the accuracy of prediction and deriving a suboptimal model [9]. The imbalanced learning problem is a major concern in the case of under-represented data and class distribution skew [10]. In large-scale optical networks, in particular, it is crucial for network operators to address the imbalanced learning problem. Undersampling and oversampling are the two main methods to solve this problem at the data level [11]. Undersampling decreases the possibility of sampling high-occurrence samples, whereas oversampling is aimed at increasing the possibility of sampling low-occurrence samples. In certain methods, no new examples are created, but the choice of samples to be replaced is informed rather than random. In other methods, such as the synthetic minority over-sampling technique (SMOTE) [12] and its variations [13–15], new examples are created from different perspectives to balance data distribution. However, these algorithms do not work when some samples are in an extremely small proportion. In these cases, other data set details need to be considered for extending the data set efficiently.

In our prior works [16,17], we had implemented an ML-enabled network platform, called Self-Optimizing Optical Network (SOON). Besides, this paper is an extension based on [18], including much extensions for SOON 2.0, refined data-processing methods, and so on. Based on the SOON 2.0, a Dirty-data-based Alarm Prediction (DAP) algorithm is proposed for addressing the dirty-data issue of data set in this paper. We also propose special data augmentation method to balance data distribution. The rest of this paper is organized as follows. Section 2 elucidates the SOON architecture. Section 3 describes the DAP algorithm procedure. Section 4 presents a large-scale network scenario and the experimental results, and Section 5 draws the conclusion.

2. SOON architecture

The Software Defined Optical Network (SDON) [19] provides a logically centralized control mechanism, flexible network function via open northbound Application Program Interface (API), etc. Based on SDON, a new network architecture called SOON [17] was proposed, which integrates ML technology with the SDON to improve the intelligent capability within optical networks. The SOON architecture is composed of three layers briefly, i.e., the data, model, and policy layers, as shown in Fig. 1(a). The data layer includes physical/virtual optical networks. The model layer is responsible for controlling the network and lifecycle of the ML models. The policy layer builds various policies for different optical network

Optics EXPRESS



(a) (b)

Fig. 1. (a) SOON architecture and (b) SOON deployment scenario.

applications, based on the trained ML models in the model layer. Each network element of data layer connects to the model layer through the management and data interface (MDI). Model layer receives instruction from the policy layer through the service and model interface (SMI).

In the data layer, the elements of the optical network support data collection in details. In the model layer, the unified network model reformats big data from the data layer into a unified data structure. The Network Service Module (NSM) provides basic services for the policy layer. In addition, the NSM provides traditional services, such as topology discovery, and ML-related services, such as model creation. The Network Control Core (NCC) and Machine Learning Engine (MLE) are two key components of the model layer. The NCC is implemented based on the Software-Defined Network (SDN) controller, and is designed to control the optical network with flexibility and scalability, but without intelligence. The MLE is implemented based on the ML platform, and is designed to train, evaluate, and apply ML models for specific network applications. Besides, the network database is specially aimed at providing efficient storage and access for the enormous data collected from the data layer. The Model Management Module (MMM) processes data from the network database to build training and testing data sets, following specific policies. The policies used in the policy layer includes application for transmission, network, and combination of them. The SOON deployment is depicted in Fig. 1(b). The data layer is carried by the backbone optical network, in different areas of China. The model layer is implemented based on the SDN controller (ONOS [20]) and distributed machine learning engine (TensorFlow [21]). ONOS and TensorFlow are the NCC and MLE, respectively. PostgreSQL [22], which stores all the training and testing data sets, is implemented as a network database module in the model layer.

Optics EXPRESS

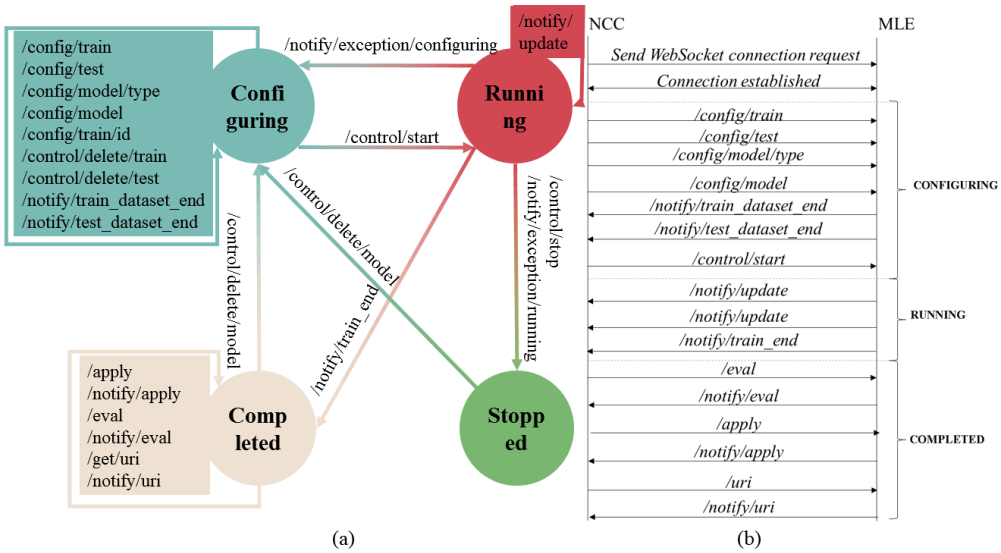


Fig. 2. SOON 2.0: (a) ML model state machine, and (b) model training and application process.

As mentioned above, in previous implementation of SOON platform [16,17], the training, testing and application for ML models are designed for different tasks. However, it's hard for SOON to handle continuous data stream for multiple network policies without a universal model management mechanism in business optical networks, especially when data need complex pre-processing. Therefore, in SOON 2.0, we develop a ML state machine in MLE to manage all events in the lifecycle of each potential ML model, for example, training/testing tasks, evaluation for trained ML models, and invokes for different models are managed in a state machine, as Fig. 2(a) shows. Then we also build a preprocessing tool as a plugin to help process collected data, which is the key to deploy the algorithm proposed in this paper. The state machine is implemented over a websocket-based protocol between ONOS and TensorFlow. There are four states for an ML model: configuring, running, stopped, and completed. In the Configuring state, the ML model can be configured with detailed parameters, including the training data set, testing data set, ML algorithm selection, and hyper-parameter settings. The Running state indicates that the ML model is running on the training data set. The Stopped state is an abnormal state indicating that an error has occurred while running, or the training process has been manually interrupted. The Completed state indicates that the ML model is ready for evaluation and application. Transition between states and the related messages for a state are realized by a websocket message, which uses the URI prefix to express the message type and data format to indicate the necessary information.

Figure 2(a) depicts all the available operations in the ML state machine. The configuration, control, and notification messages are three types of messages with "*/config/*", "*/control/*", and "*/notify/*" prefixes, respectively. The "*/apply*", "*/eval*", and "*/uri*" messages request application, evaluation, and the TensorBoard page of the trained ML model. The model configuration message configures an ML model from the NCC to MLE, whereas the model control message controls active state transition. Model notification notifies the execution results and exceptions, and also indicates state transition, when exception occurs or training ends. For an ML model, the normal state transition route includes configuring, running, and completed (see Fig. 2(b)). In the Configuring state, the NCC sends the data set to the MLE after preprocessing the original data from the database, and configures all the parameters. The NCC then sends a "*/control/start*" message to start training the MLE, and the state is set as Running. In Running state, "*/notify/update*" messages are sent to the NCC for notifying the training process. After training, Completed state is reached through the

"*/notify/train_end*" message. In this state, the trained model is used for performance evaluation and practical application. Besides, the NCC also obtains the TensorBoard link for operators, which is a webpage that shows the intermediate results during training. These results are often used for analyzing the training process for further optimization. The Stopped state occurs, if training fails or the operator is forced to stop. The "*/control/delete/model*" message can revert the state back to Configuring state by deleting the uncompleted model.

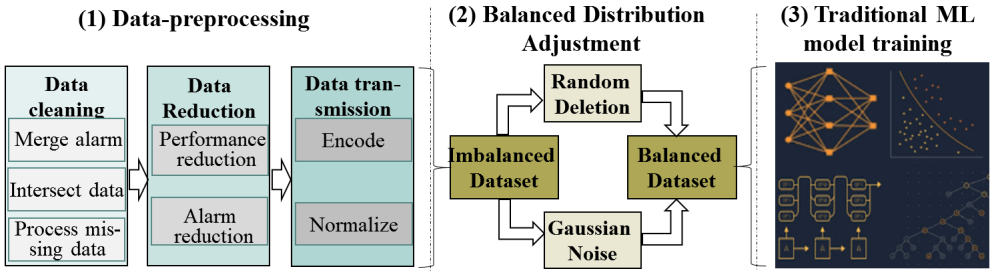


Fig. 3. DAP procedure.

3. Dirty-data-based alarm prediction

As previously mentioned, the alarm data collected from the model layer are dirty. Therefore, based on the SOON 2.0, we propose a Dirty-data-based Alarm Prediction (DAP) algorithm to handle dirty data and train the ML models. This algorithm involves three steps: data preprocessing, balanced distribution adjustment, and traditional ML model training, as shown in Fig. 3.

3.1 Data preprocessing

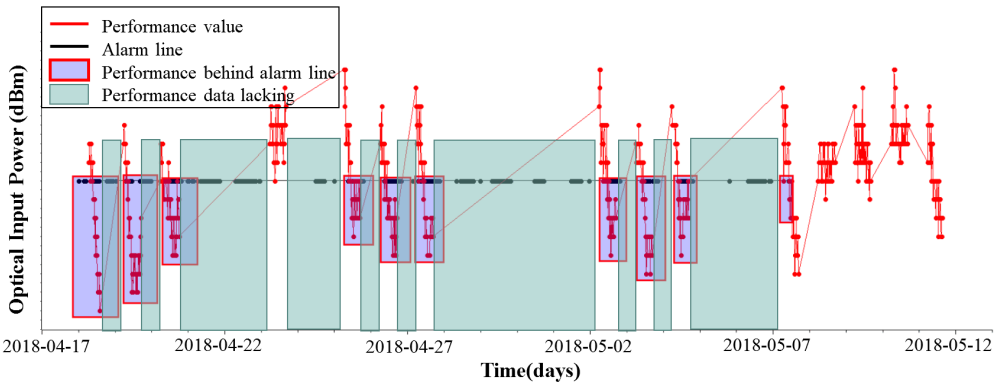


Fig. 4. Demonstration of collected performance data.

Real-world data is often incomplete, inconsistent, and likely to contain many errors. The results obtained by directly analyzing dirty real-world data are unreliable. Hence, data preprocessing is a necessary step for mining the data set from dirty data. Data cleaning, data integration, data reduction, and data transmission are the four major steps in data preprocessing [23]. Data cleaning fills in the missing values, smooths noisy data, and identifies the outliers. Data integration merges data from multiple data sources. Data reduction is applied to obtain a considerably smaller representation of the original data, which maintains similar characteristics. Data transmission includes smoothing, attribute construction, aggregation, encoding, normalization, discretization, and concept hierarchy generation for nominal data. There are lots of general methods in these steps for all kinds of data, however, there are still some extra special methods for specific application. In Fig. 3,

Optics EXPRESS

merge alarm in data cleaning phase and *alarm reduction* in data reduction phase are designed only for failure prediction according to the characteristics of performance data and alarm data.

Practical data preprocessing with various original data is different, depending on the data quality, target requirement, application domain, etc. In our study, we use the PostgreSQL database to collect massive performance and alarm data from 274 optical equipment. Figure 4 shows the optical input power in an optical node from 17th April to 12th May. The red line indicates the performance value and the black indicates the alarm line. If the value is lower than the alarm threshold, an alarm will be reported. The black spots in the black line indicate the timestamp of the alarm event, called IN_PWR_LOW. The Current alarm and historical alarm are two types of alarm records. An alarm record is considered as a current alarm, once it is reported. When this alarm is confirmed and cleared by the operator, it is converted to a history alarm record. Current alarms contain only the alarm frequency, first occurrence time, and last occurrence time 'within a period of time', and the exact timestamp of the event is unknown. Alarm frequency means times that specific alarm happened in a period. If an alarm happens but has not been confirmed and cleaned, it will be included in alarm frequency. Hence, current alarm information cannot be used directly for alarm prediction. In Fig. 4, the light cyan blocks indicate the blank periods, when no performance data were collected, and the light purple blocks indicate the time intervals, when performance value is lower than the alarm line.

When inserting data into the database, duplicated data collected twice, and conflicted data at the same time point are skipped because of the primary key constraint. Further, the data cleaning, data reduction, and data transmission methods are used for preprocessing the original data. Our data cleaning process involves three steps:

1. Merge the current and history alarms. If the alarm frequency of a current alarm is two, the time of the first and last occurrences are the event times of these two alarms, respectively. Then, some of the current alarms could be converted to 2x history alarms and the others would be skipped.

2. Intersect the performance and alarm data. The collected performance and alarm data should be matched in the same time period, for building the alarm prediction data set. For example, as depicted in Fig. 4, the prediction data set from 22nd April cannot be built because there are some alarms but no performance data on this day.

3. Process missing data. The sampling interval for the performance data is 15 minutes. We use the median value of two adjacent values to fill a single missing data. If more than one data point are missing in time series, this missing item is removed because overmuch information lacking makes it unavailable.

The next step is data reduction. In optical networks, the performance values remain unchanged or change very slightly, most of the time. Therefore, most of the performance values are not useful for alarm prediction. In order to reduce the volume of original data, we merge the continuous unchanged values into one data item. Besides, as shown in Fig. 4, when the performance value is lower than the alarm line, alarms will be triggered continuously. Hence, only the first alarm is useful, and all the subsequent alarms are unnecessary, and would be removed, when building the data set.

Data transmission is the last step in data processing. In this paper, the data transmission is easy to do. Because smoothing will destroy the data distribution of performance data. The used attribute is selected in time series, which doesn't need further construction. There is no obvious ranking in performance data, so we don't use discretization. The two things we complete in data transmission phase are encoding and nominalization. Encoding maps data into float-point numbers, and normalization shifts performance values and scale them in scope [0, 1].

Optics EXPRESS

3.2 Balanced distribution adjustment

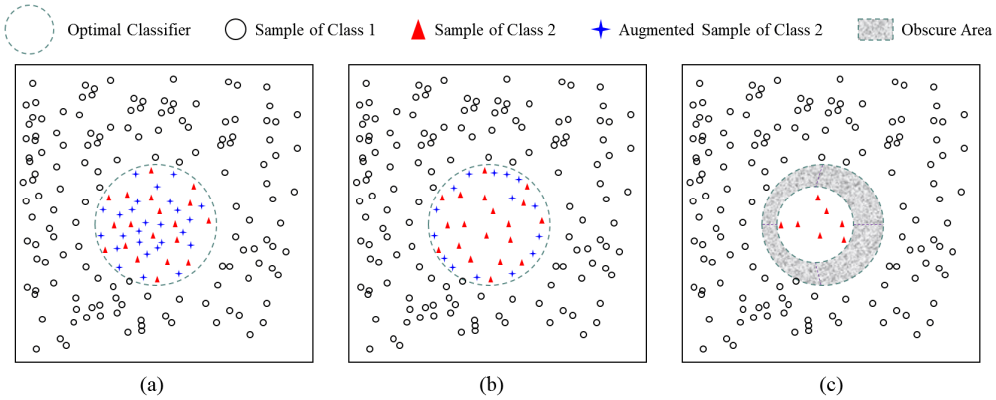


Fig. 5. Binary classification cases, (a) SMOTE data augmentation, (b) borderline-SMOTE data augmentation, and (c) scenarios where traditional data augmentation methods do not work.

After specific data preprocessing, an available data set is built. In a network alarm data set, whose inputs include time-series correlated performance data, the output indicates whether an alarm will occur in future. If alarm data caused by a specific reason play a major role in the data set, it is easy for the ML model to realize high-precision for predicting this alarm, but it becomes difficult to predict the other alarms efficiently. Because traditional ML model will focus on feature learning of the major alarm based on prejudiced data set. Hence, unbalanced data distribution for a specific target influences the algorithm performance. Actually, the data with unbalanced attribute could be considered as a kind of dirty data. Many traditional methods have been proposed for solving this problem. Figure 5 shows two classical data augmentation algorithms, SMOTE [12] and Borderline-SMOTE [13], for a binary classification case with 2-dimensional data. The class-2 sample number is relatively small, and the dashed circle indicates the optimal classifier that splits two classes. The SMOTE uses k nearest neighbors (KNN) to determine the similar neighbors of a sample, and generates a new synthetic sample using the difference between the sample and its neighbors. The blue crosshairs in Fig. 5(a) are the augmented samples using SMOTE, and are located within the inner dashed circle. Based on the SMOTE, the Borderline-SMOTE focuses on generating new samples located near the boundary because these samples can contribute more toward the demarcation of the classification boundary. Gray zones always exist between different classes, as shown in Fig. 5(c). The evaluation of the gray zone is generally very difficult. In this case, the introduction of domain knowledge is an available method for rendering the boundary more distinguishable. Because sometimes ML algorithm couldn't extract information when this information is not included, or hidden very deeply in train data set. The

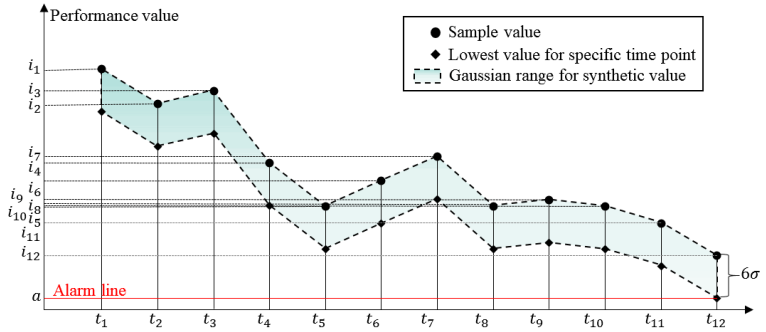


Fig. 6. Data augmentation of limited Gaussian noise.

Optics EXPRESS

introduction of domain knowledge could help train data set expose more features obviously, which is easy to learn.

In order to balance the data set for mean distribution by changing the data size of each alarm, we use random deletion to decrease the data volume, and limited Gaussian noise (LGN) to increase the data volume for the alarms. Random deletion removes data randomly, and the LGN is used for generating new trusted data without changing the key features. In optical equipment, alarms are mostly triggered when specific related and monitored performance value reaches a special threshold and keep it for a duration, or in other conditions. The triggering condition can be used to extend data. For example, the IN_PWR_LOW alarm is triggered, when the optical input power is lower than the alarm line. In Fig. 6, we assume that $I = \{i_1, i_2, ..., i_{12}\}$ is the input in the time dimension, which is the main part of ML algorithm's input (see section 3.3). $\min(I)$ is the minimal value of I, and a is the alarm line. 6σ is the variation interval of original sample value. The *k*-th value of a new synthetic sample couldn't be out of scope $[i_k - 3\sigma, i_k + 3\sigma]$, which actually comes from three-sigma rule of Gaussian distribution. This scope is a confidence interval for such distribution, and is proved by thousands of facts. The probability of value x at time t_k, is given by Eq. (1). p_k follows the Gaussian distribution, where the mean is $i_k - (\min(I) - a)/2$, and variance is $(\min(I) - a)/6$. The LGN could generate new synthetic samples in the gray zone by modify i_k one by one. Because it introduces external knowledge into data augmentation, instead of only retrieving features within the original data.

$$p_k = \frac{6}{\sqrt{2\pi}(\min(I) - a)} \exp\left(-\frac{9(2x - 2i_k + \min(I) - a)^2}{2(\min(I) - a)^2}\right) \quad (1)$$

3.3 Traditional ML model training

Based on the balanced data set, traditional ML algorithm driven classification can be used to predict network alarms. In this paper, we actually compare several classical ML algorithms under different hyper-parameters. The result shows that all algorithms with proper hyper-parameters reach almost the same accuracy rate for alarm prediction in this paper, which means these ML algorithms are able to learn the features of data set. And the key point is the quality improvement for data set. Although some research works [24] compare multiple algorithms for their problem, the detailed study for different ML algorithms is beyond the scope of this paper, so we only choose artificial neural network (ANN), and focus on detailed description on this algorithm.

Optics EXPRESS

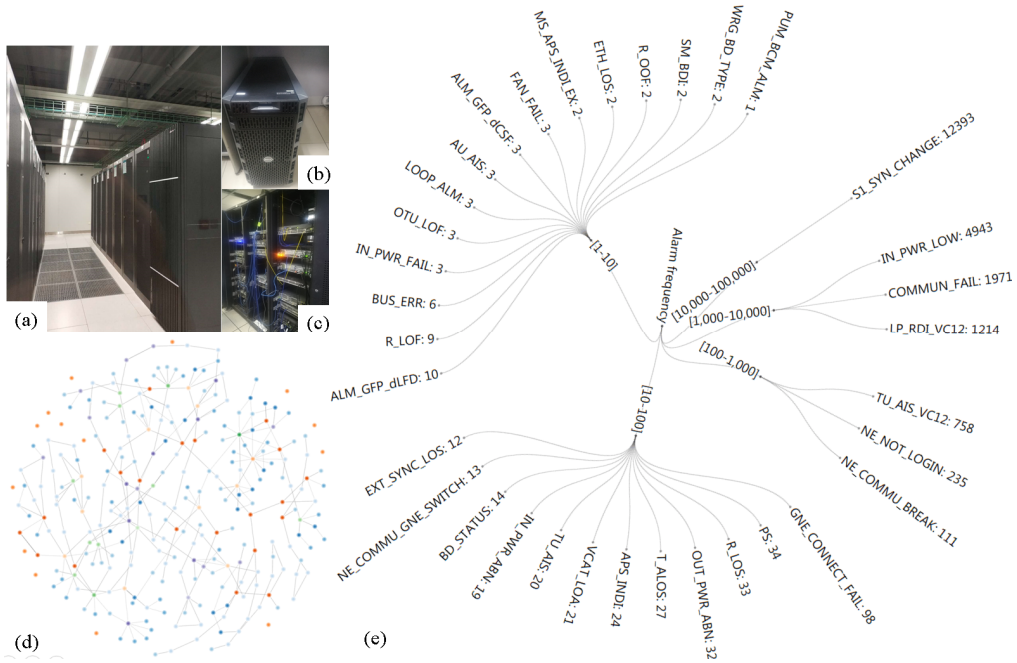


Fig. 7. Experimental scenario: (a) Telecommunications room, (b) GPU server, (c) computing server and storage server, (d) large-scale field network with 274 nodes and 487 links, and (e) history-alarm distribution.

ANN is a classical multi-layer ML algorithm including input layer, hidden layer, and output layer. Each layer contains different amounts of neurons. Each neuron contains an activation function with adjustable parameters, for example, sigmoid function and rectified linear unit. Neuron use activation function to calculate the output value for its input value. There are one-to-one connections between each neuron of the layer and each neuron of its adjacent layer, which means each neuron would gather all information from previous layer, and extract high-level information. Therefore, ANN is able to express non-linear complex mapping between two vectors. Input layer is the entrance of ANN, and carries a vector as input. Output layer is the exit of ANN, and shows a vector as output. Hidden layers are responsible to extract abstract information to find a proper mapping between input and output. The parameters in each neurons are not constant, which will be updated according to loss value. Loss is the difference between output of train data set and output of ANN model. According to loss value, gradient of all parameters in each neuron could be calculated, and used to update parameters via learning rate. After thousands of updates under proper configurations, ANN model would perform better and better, and reach high accuracy rate finally.

4. Field-trial scenario and results

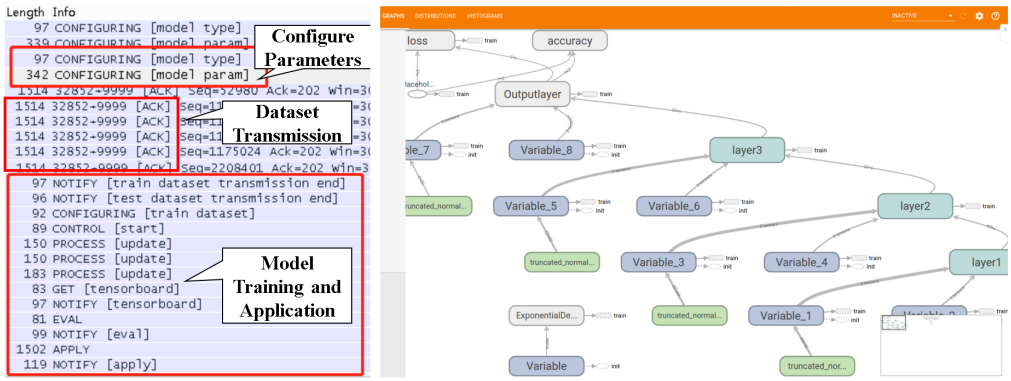


Fig. 8. Experimental testbed: (a) State machine captured by Wireshark and (b) TensorBoard webpage.

4.1 Description for field-trial scenario

We implemented data preprocessing in the unified network model (UNM), and balanced distribution adjustment and ANN in the MLE. The alarm prediction policy in the policy layer controls the above mentioned three steps via SMI. The training data set was collected from large-scale commercial optical networks. Figure 7 shows the field-trial scenario, including the telecommunication Central Office (CO) of the network operator (Fig. 7(a)) and the servers (Fig. 7(b) and Fig. 7(c)), which include the GPU, computer, and storage units. The MLE, NCC, and network database run on these servers. 56,901 history alarms, 3,124 current alarms, and 11,207,971 performance items were collected from April 18th to May 13th, 2018, from a large-scale optical network with 274 nodes and 487 links, as depicted in Fig. 7(d). The large-scale field network is the backbone SDH optical network in Chinese mainland including 27 provinces. This network contains 73 devices of OptiX OSN 3500, 81 devices of OptiX OSN 7500, 2 devices of Metro 1000v3, 10 devices of OptiX 2500 + , 1 device of OptiX 1500, 3 devices of Metro 100, and 104 devices of other types. Figure 7(e) shows the alarm frequency distribution of the history alarms. We observed that the alarm distribution was extremely unbalanced in a production environment. The S1_SYN_CHANGE alarm constitutes 21.7799% of the history alarms, whereas the PUM_BCM_ALM alarm is only 0.0018%; the former is 12393 times the later. We selected IN_PWR_LOW, OUT_PWR_ABN, and R_LOS as the predicted alarms. Then, according to correlation analysis, we used the optical input power, pump laser operating current, and optical output power as the input respectively.

4.2 ANN model formulation

The original ratio of these three alarms was 98.7:0.64:0.66. After preprocessing, the ratio became 84.13:14.29:1.59, which was still unbalanced. At the moment, we got original data set. The input $\boldsymbol{X}$ of the data set is a 1-dimension vector that is consisting of 13 single-precision floating-point numbers, as Eq. (2) shows. x_1 is set as one of $\{0,0.5,1\}$, which means three alarm type selected from all alarms respectively, as Eq. (3) shows. And the other 12 values indicate performance in time series. The 12th value means the performance 15 minutes ago. The time period is 15 minutes, so the 1st value means the performance 3 hour ago. The output y contains only one value, which is the probability of an alarm event within the next 15 minutes. The scope of y is $[0,1]$. th is a threshold to confirm whether failure would happen in 15 minutes. As Eq. (4) shows, if y is larger than, it means there should be failure in 15 minutes. Otherwise, the network maintains normal work in future 15 minutes.

Optics EXPRESS

The value of *th* has important influence on the judgement of failure. In this simulation, *th* is set as 0.5.

In order to evaluate the performance of the DAP, we used a simple ANN algorithm, without a balanced distribution adjustment process, as the benchmark. We extended the alarms such that the sizes of the three alarms were the same, and selected a quarter as the test data set, randomly. Further, we selected 25%, 50%, 75%, and 100% of the remaining three-quarter alarms as the training data sets of DAP (25%), DAP (50%), DAP (75%), and DAP (100%), respectively. The DAP as well as ANN use three hidden layers with 64, 64, and 96 neurons, respectively. Our simulation shows that the performance of ANN with more hidden layers is unstable, that means ANN is easy to be overfitting. Also, we compare different hyper-parameters of same ANN, and find that the performance is the best and most stable when learning rate is 5e-3 in scope [1e-1, 1e-5], learning rate decay policy is constant in option set {constant, exponential, piecewise constant}, and batch size is 4 in option set {1, 2, 4, 8, 16, 32}. There are lots of detailed comparisons for ANN algorithm, which is not the point focused in this paper. So only the best combination is selected to show. Besides, the mean square error (MSE) loss function, stochastic gradient descent [25] optimizer, and the other more detailed parameters are used.

$$X = (\underbrace{x_1}_{\text{Type value}}, \underbrace{x_2, x_3, \ldots, x_{12}, x_{13}}_{\text{Performance value}}) \tag{2}$$

$$x_1 = \begin{cases} 0, & \text{IN\_PWR\_LOW} \\ 0.5, & \text{OUT\_PWR\_ABN} \\ 1, & \text{R\_LOS} \end{cases} \tag{3}$$

$$y \in \begin{cases} [0, th), & \text{No failure} \\ [th, 1], & \text{Failure} \end{cases} \tag{4}$$

4.3 Results

Figure 8 displays the SOON 2.0 experimental testbed. Figure 8(a) shows the state change of the DAP and ANN models captured by the Wireshark tool developed by our team. After configuring the model parameters and data set transmission, the state changes to Running from Configuring. After model training is completed, */uri*, */eval*, and */apply* messages are sent

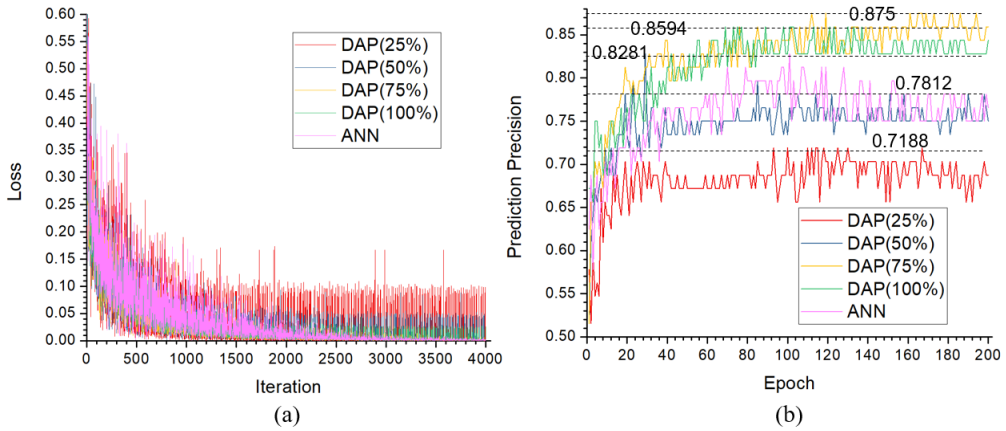


Fig. 9. Experimental results: (a) loss variation with iteration and (b) prediction precision.

Optics EXPRESS

to the MLE from the NCC for evaluating and applying the trained model. Figure 8(b) depicts the TensorBoard webpage, which contains the detailed model configuration and training procedure for further optimization.

Figure 9(a) shows the loss variation, when the iteration increases for the DAP and ANN algorithms. Loss value is calculated by MSE, which is a metric to evaluate the gap between current model and ideal optimal model on this data set. Generally, lower loss value means better performance on train data set. However, a model that performs well on train data set may performs poor on test data set, which is called overfitting. So low value is the necessary condition but not sufficient condition to check if a model is optimal. In Fig. 9(a), all the five algorithms converged to less than 0.1, after approximately 3800 iterations. The loss of DAP (25%) was relatively high because the size of its training data set is the least. Figure 9(b) shows the prediction precision. Compared to the ANN, whose highest precision is 82.81%, the DAP (50%) and DAP (75%) are higher, with 85.94% and 87.50% precision, respectively. The performance improvement is 4.69% at most between the ANN and DAP (75%). This indicates that balanced data is learnt by the ML algorithm rapidly and accurately. However, the DAP (25%) and DAP (100%) are relatively low because many features in the DAP (25%) were removed by random deletion, and DAP (100%) introduces excessive Gaussian noise.

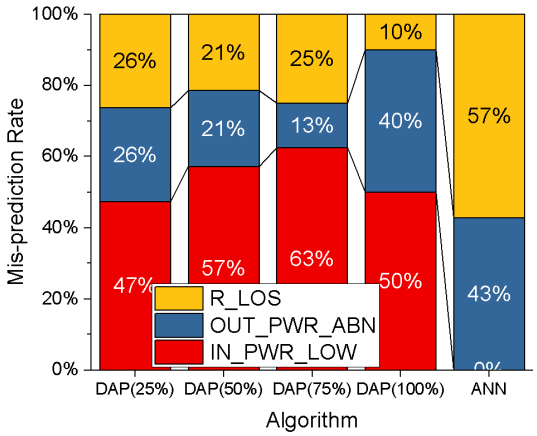


Fig. 10. Prediction error analysis.

Figure 10 further analyzes the reason for the performance improvement of the DAP. IN_PWR_LOW was not mis-predicted in the ANN, indicating that it learns the features of this alarm perfectly on an unbalanced data set, but skips the feature learning of the other alarms. In the DAP, the learning for each type of alarm was relatively balanced. The mis-prediction rates of R_LOS and OUT_PWR_ABN in the DAP (25%–100%) were unstable because these samples were augmented and disturbed by Gaussian noise. However, the size relationship of IN_PWR_LOW in the DAP (25%–100%) was the same as the precision-size relationship in Fig. 9(b), i.e., *DAP (75%) > DAP (50%) > DAP (100%) > DAP (25%)*. This indicates that the sum of the mis-prediction rates of R_LOS and OUT_PWR_ABN increases, when the prediction precision decreases, further proving that data augmentation enables the model to learn features in a more balanced way.

5. Conclusions

In this paper, we introduced SOON architecture for integrating ML technology into the control and management of large-scale commercial optical networks. In order to make SOON more suitable for failure prediction, we designed a websocket-based protocol to manage the state machine of ML models, and developed a pre-processing tool. Based on the SOON, we proposed a DAP algorithm for processing dirty data and overcoming the influence of

Optics EXPRESS

imbalance distribution in a data set. Experimental results demonstrated that DAP can improve the alarm prediction accuracy by 4.69%, compared to the traditional approach.

Funding

China State Grid Corp Science and Technology Project (5210ED180047), National Science and Technology Major Project (2017ZX03001016), National Natural Science Foundation of China (NSFC) (61822105, 61571058, 61601052), State Key Laboratory of Advanced Optical Communication Systems Networks of China.

References

1. Y. R. Zhou, K. Smith, M. Gilson, J. Chen, W. Pan, Y. Chang, S. Wu, S. Wu, and I. Davis, "Demonstration of real-time 400G single-carrier ultra-efficient 1.2 Tb/s superchannel over large Aeff ultra-low loss terrestrial fiber of 150 km single span and 250 km (2×125 km spans) using only EDFA amplification," in Optical Fiber Communication Conference, OSA Technical Digest (online) (Optical Society of America, 2018), paper M1E.4.
2. M. Mazur, A. Lorences-Riesgo, J. Schroder, P. A. Andrekson, and M. Karlsson, "10 Tb/s PM-64QAM self-homodyne comb-based superchannel transmission with 4% shared pilot tone overhead," J. Lightwave Technol. **36**(16), 3176–3184 (2018).
3. S. Rahman, T. Ahmed, M. Huynh, M. Tornatore, and B. Mukherjee, "Auto-scaling VNFs using machine learning to improve qos and reduce cost," in Proc., IEEE Intl. Conf. on Commun., Kansas City, USA, 2018, pp. 1–6.
4. B. Yan, Y. Zhao, Y. Li, X. Yu, J. Zhang, Y. Wang, L. Yan, and S. Rahman, "Actor-critic-based resource allocation for multi-modal optical networks," in Proc. IEEE GLOBCOM Workshop on Machine Learning for Communications, Abu Dhabi, UAE, Dec. 2018.
5. Z. Wang, M. Zhang, D. Wang, C. Song, M. Liu, J. Li, L. Lou, and Z. Liu, "Failure prediction using machine learning and time series in optical network," Opt. Express **25**(16), 18553–18565 (2017).
6. L. Barletta, A. Giusti, C. A. Rottondi, and M. Tornatore, "QoT estimation for unestablished lighpaths using machine learning," in OFC, Th1J.1, 2017.
7. T. Panayiotou, S. P. Chatzis, and G. Ellinas, "Leveraging statistical machine learning to address failure localization in optical networks," J. Opt. Commun. Netw. **10**(3), 162–173 (2018).
8. W. Kim, B. Choi, E. Hong, S. Kim, and D. Lee, "A taxonomy of dirty data," Data Min. Knowl. Discov. **7**(1), 81–99 (2003).
9. Q. Wang, Z. Luo, J. Huang, Y. Feng, and Z. Liu, "A Novel Ensemble Method for Imbalanced Data Learning: Bagging of Extrapolation-SMOTE SVM," Comput. Intell. Neurosci. **2017**, 1827016 (2017).
10. H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE T Knowl. Data En. **21**(9), 1263–1284 (2009).
11. N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," SIGKDD Explor. **6**(1), 1–6 (2004).
12. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," J. Artif. Intell. Res. **16**, 321–357 (2002).
13. H. Han, W. Wang, and B. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in International Conference on Intelligent Computing (Springer 2005), pp. 878–887.
14. Y. Mi, "Imbalanced classification based on active learning SMOTE," J Appl. Sci. En. Technol. **5**(3), 944–949 (2013).
15. G. Douzas and F. Bação, "Geometric SMOTE: Effective oversampling for imbalanced learning through a geometric extension of SMOTE," arXiv 1709.07377 (2017).
16. B. Yan, Y. Zhao, W. Wang, L. Yan, Y. Wang, J. Liu, S. Zhang, D. Liu, Y. Lin, H. Zheng, and J. Zhang, "First demonstration of machine-learning-based self-optimizing optical networks (SOON) running on commercial equipment, " in European Conference on Optical Communication, Roma, Italy, 23–27 Sept. TuDS.3, 2018.
17. Y. Zhao, B. Yan, D. Liu, Y. He, D. Wang, and J. Zhang, "SOON: self-optimizing optical networks with machine learning," Opt. Express **26**(22), 28713–28726 (2018).
18. B. Yan, Y. Zhao, Y. Li, X. Yu, J. Zhang, and H. Z. Yilin, "First Demonstration of Imbalanced Data Learning-Based Failure Prediction in Self-Optimizing Optical Networks with Large Scale Field Topology," in 2018 Asia Communications and Photonics Conference (ACP), Hangzhou, China, pp. 1–4, 2018.
19. X. Zheng and N. Hua, "achieving inter-connection in multi-domain heterogeneous optical network: from PCE to SDON," in Asia Communications and Photonics Conference 2013, OSA Technical Digest (online) (Optical Society of America, 2013), paper AW4I.2.
20. Open Networking Foundation, "Open Networking Operation System," *https://onosproject.org/*.
21. Google Inc, "TensorFlow," *https://www.tensorflow.org/*.
22. The PostgreSQL Global Development Group, "PostgreSQL: The world's most advanced open source relational database," *https://www.postgresql.org/*.
23. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques* (Elsevier, 2011), Chap. 3.
24. R. M. Morais and J. Pedro, "Machine Learning Models for Estimating Quality of Transmission in DWDM Networks," J. Opt. Commun. Netw. **10**(10), D84–D99 (2018).
25. H. Robbins and S. Monroe, "A stochastic approximation method," Ann. Math. Stat. **22**(3), 400–407 (1951).