Auxiliary Fault Location on Commercial Equipment Based on Supervised Machine Learning

Abstract: As the fundamental infrastructure of the Internet, the optical network carries a great amount of Internet traffic. There would be great financial losses if some faults happen. Therefore, fault location is very important for the operation and maintenance in optical networks. Due to complex relationships among each network element in topology level, each board in network element level, and each component in board level, the concrete fault location is hard for traditional method. In recent years, machine learning, especially deep learning, has been applied to many complex problems, because machine learning can find potential non-linear mapping from some inputs to the output. In this paper, we introduce supervised machine learning to propose a complete process for fault location. Firstly, we use data preprocessing, data annotation, and data augmentation in order to process original collected data to build a high-quality dataset. Then, two machine learning algorithms (convolutional neural networks and deep neural networks) are applied on the dataset. The evaluation on commercial optical networks shows that this process helps improve the quality of dataset, and two algorithms perform well on fault location.

Keywords: optical network; fault location; supervised machine learning

ZHAO Zipiao[1], ZHAO Yongli[1], YAN Boyuan[1], WANG Dajiang[2]

(1. State Key Laboratory of Information Photonics and Optical Communictions, Beijing University of Posts and Telecomunications, Beijing 100876, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.2022S1002

http://kns.cnki.net/kcms/detail/34.1294.TN.20220106.1424.002.html, published online January 7, 2022

Manuscript received: 2020-04-01

Citation (IEEE Format): Z. P. Zhao, Y. L. Zhao, B. Y. Yan, et al., "Auxiliary fault location on commercial equipment based on supervised machine learning," *ZTE Communications*, vol. 20, no. S1, pp. 07 - 15, Jan. 2022. doi: 10.12142/ZTECOM.2022S1002.

1 Introduction

With the explosive growth of Internet traffic and the emergence of new applications, the requirement for traffic bandwidth increases rapidly. As the fundamental infrastructure of the network, the optical network becomes more and more important because of its super-high transmission speed. Therefore, there should be considerable economic damages in case of optical network failures. In reality, the economic loss is closely related to the network restoration time, which consists of the time to locate the fault, to drive to the place, and to fix the problem. If the fault location is fuzzy and even wrong, the restoration time will be doubled and redoubled. Hence, it's important to find the fault location rapidly and accurately[1].

Until now, some researchers have studied fault location from different aspects. Generally, there is no protocol to detect network faults in transparent optical networks, because such protocols will cost more in equipment manufacture and increase the complexity of current protocol stacks. The current fault location technology is inefficient, which depends on the expert knowledge base and traditional fiber link fault location. However, the breakthrough of machine learning in recent years offers the possibility of solving this problem.

Machine learning (ML) is an important subset of artificial intelligence, which builds a mathematical model based on sample data, known as "training data", to make predictions or decisions without being explicitly programmed to perform the task[2]. The introduction of ML into fault location has become a hot spot. In Ref. [3], a recurrent neural network-based method is used in the fault detection and isolation of the satellite's attitude control subsystem. In Ref. [4], a fault section estimation in electric power systems is handled, where neural networks are employed to model the protection systems, dealing with the uncertainties involved with relay and circuit-breaker operation messages. In Ref. [5], a deep

This work was supported in part by National Natural Science Foundation of China (NSFC) (61822105, 61571058, 61601052) and by ZTE Industry-Academia-Research Cooperation Funds under Grant No. 2018110016001047.

learning (DL) based algorithm is proposed to combine supervised and unsupervised learning, and solve the fault in power systems. Also, there are some research works related to the fault in optical networks[6]. In Refs. [7 – 9], a Bayesian inference method is used to judge service-triggered failure in optical transport networks and self-diagnosis in fiber to the home (FTTH) gigabit-capable passive optical networks (GPON). In Ref. [10], the previously established lightpaths with already available failure localization and monitoring data are used to estimate the failure localization at the link level for all lightpaths with Kriging method. In Ref. [11], two ML based algorithms are described based on regression, classification, and anomaly detection. The authors propose a bit error ratio (BER) anomaly detection algorithm which takes historical information (like maximum BER, threshold BER at set-up, and monitored BER per lightpath) as input, and detects any abrupt change in BER which might be a result of some failures of components along a lightpath. In Ref. [12], two algorithms for testing optical switching at connection setup time and failure rootcause localization for optical network are proposed using decision tree and support vector machine (SVM) algorithms. And a similar multi-ML algorithm for failure detection and classification is proposed in Refs. [13 – 14], including binary and multi-class SVMs, random forests and neural networks (NN). However, the work mentioned above fails to take full advantage of the huge data collected from optical networks, and is difficult to be deployed into the real commercial environment. In this paper, we study the alarm data preprocessing to improve the quality of the dataset and mine the association rules. Based on the dataset, two fault location algorithms based on the convolutional neural network (CNN) and the deep neural network (DNN) respectively are proposed.

The rest of this paper is organized as follows. In Section 2, the originally collected data are converted into the universal format, and some association rules are studied. Then the mappings between the input and the label are built for an available dataset. Meanwhile, some data augmentation methods are used to expand the size of the dataset. In Section 3, CNN and DNN algorithms are explained and introduced to help solve the fault location problem. In Section 4, based on the original data collected from commercial optical networks, the procedure described in Section 2 and the algorithm are evaluated with the Tensorflow[15] AI platform. In Section 5, we draw a conclusion.

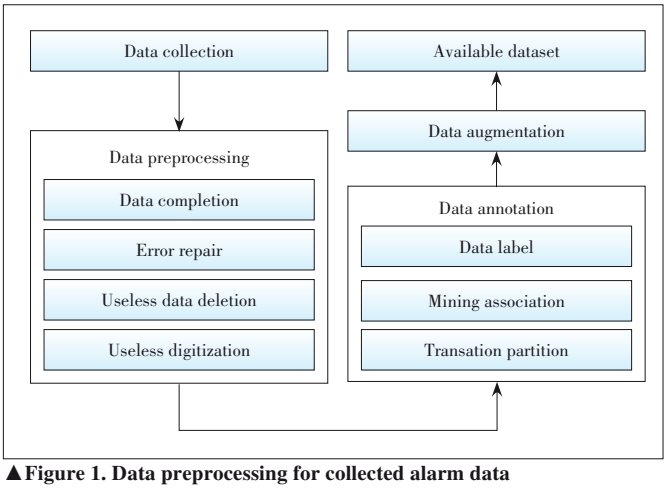
2 Building Alarm Dataset

The quality of a dataset has an important influence on the performance of fault location, which determines the performance limit directly and impacts model training indirectly. Therefore, it is essential to build a high-quality dataset based on original data. As Fig. 1 shows, in order to make an available dataset, data collection, data preprocessing, data annotation, and data augmentation are executed successively.

There are generally four kinds of data in optical equipment: the performance data, the alarm data, the configuration data, and the event report. All these data could be collected from the equipment management system directly. In the phase of data collection, we use a self-optimizing optical network (SOON) platform[16 – 17] to gather all data through an open application programming interface (API) of optical equipment.

The original data collected from optical equipment directly are generally dirty[18], which means that these data include missing data, wrong data, non-standard representations of the same data, etc.[19]. In a fault location application, we only need to process alarm data, including history alarm data and current alarm data. The history alarm data are the alarms that have been confirmed and cleaned. And the current alarm data summarize the happened alarms that have not been confirmed and cleaned simultaneously. The indices of history alarm data include the alarm level, alarm name, alarm source, network element type, location information, happen time, clean time, and confirm time. And the indices of current alarm data include alarm level, alarm name, alarm source, location information, frequency, time to first alarm, time to last alarm, and clean time. In the phase of data preprocessing, data completion is used to remove the severely defective data records and complement partial missing data records. Error repair is used to detect obvious data errors in format and content, and fix them. Then, useless data deletion is used to remove the useless data records. For example, if the repair for the network failure is delayed for a while, the same alarm data will be triggered again and again. In hundreds of the same alarm records, we only need one of them. Finally, we digitalize all data by mapping the text to a specific integer.

After some steps of data preprocessing, high-quality alarm data are generated. Because the used algorithms are super-



▲Figure 1. Data preprocessing for collected alarm data

vised, we need to mark data to add a label for each data item. Firstly, we use a sliding window[20] to convert alarm data records into transactions. We assume that b is the beginning time, w is the length of a time window, and s is the step length between two adjacent transactions. Then the k-th transaction includes the data records in time period $(b+k*s, b+k*s+w]$.

According to different value setups of b, s, and w, we could get multiple original datasets. Considering known algorithms to mine association rules, we select a frequent pattern growth[21] (FP-growth) algorithm, as it is one of the fastest approaches to a frequent item set mining[22]. The key idea of FP-growth is partition, which executes a frequent pattern tree (FP-tree) to generate a common frequent pattern through only two times that of data traversal. The process is as follows.

1) Firstly, it scans all stored alarm transactions, calculates the frequency of each transaction as the support degree, and drops those transactions whose support degree is less than the minimal support degree. Then, it sorts all transactions with support degrees in descending order, and stores them in a list L.

2) Secondly, an FP-tree is built. In an FP-tree, the root node is initialized as null. Then, a frequent pattern subtree is created. If the conditional pattern bases of FP-tree are the same, the support value on these nodes is plus 1. If not, then it creates a new node with the support degree as 1.

3) Lastly, it mines FP-tree in a reverse order of list L, creates the conditional pattern base for each item, and combines all bases for the same alarm data to build its FP-tree. This step is repeated until there is no unused conditional pattern base.

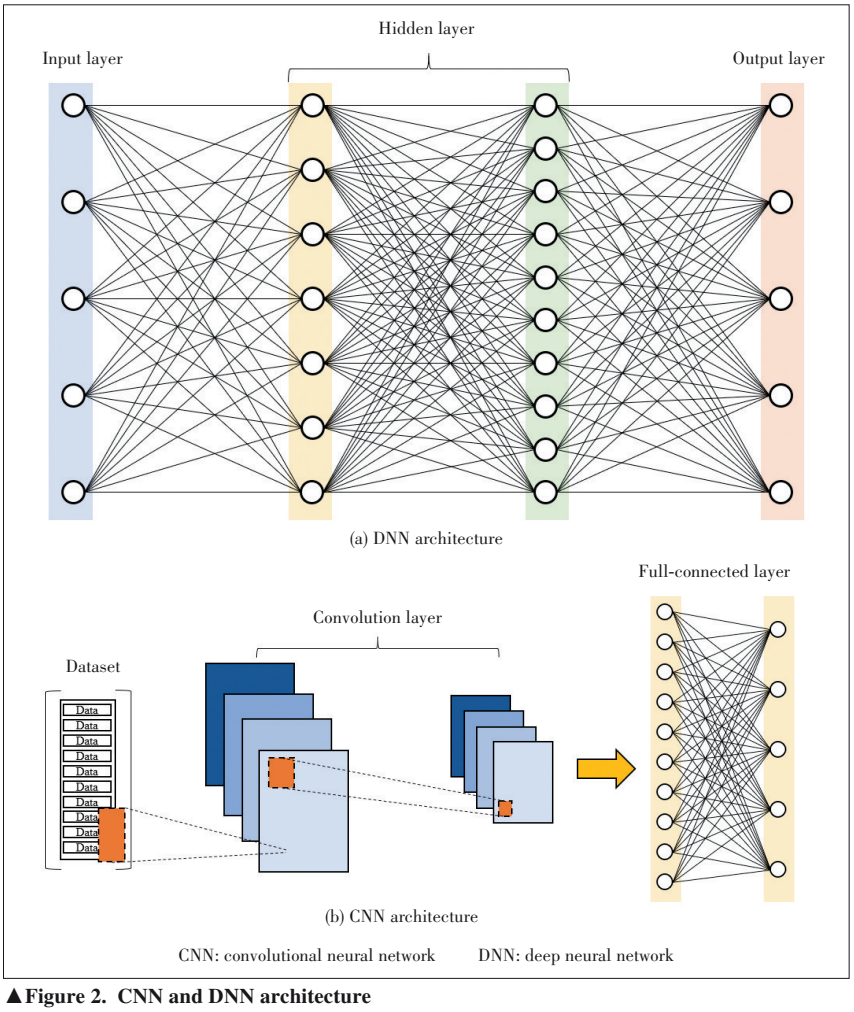
The last phase to generate an available dataset is to augment data. In a commercial optical network, the data about network failures are much less than those in the normal state. So, the size of a fault dataset is quite small. Meanwhile, the distribution for events caused by different reasons is uneven. In order to avoid negative effects of small size and maldistribution, data augmentation could be used to expand specific data and keep the distribution balance. In image recognition, an image could be augmented to multiple images through rotation, translation, reversal, clipping, Gaussian noise, etc. In fault location, although there are already several data augmentation methods[17], we still need to manually expand data depending on artificial expertise under specific scenarios. For example, if we want to expand alarm data caused by clock failure, firstly we should abstract all clock topologies that exist in an optical network, and mark where the clock is for each clock topology. For alarm data caused on a specific clock, we could simulate the same failure on other clock to generate new clock alarm data.

3 Auxiliary Fault Location

CNN and DNN are two important tools in supervised machine learning which aim to find an accurate mapping between the input and the output. Based on the available dataset generated in Section 2, these two algorithms are used to find the relationship between the alarm data and the fault location.

3.1 Fault Location with Deep Neural Network

DNN is actually the fully-connected neural network where the number of layers is larger than four. Based on huge labeled training data, DNN can find the proper relationship mapping between the input and the output, especially for those fuzzy and complex mappings. As Fig. 2(a) shows, DNN



▲Figure 2. CNN and DNN architecture

consists of the input layer, hidden layer, and output layer. Each layer contains multiple neurons, and each neuron could be defined as a non-linear function (Sigmoid[23], tanh, Rectified Linear Unit[24], etc.). Compared with the neural network with only 3 layers, DNN owns stronger feature expression. The update of DNN model normally consists of forward and back propagation. The forward propagation calculates the output of each neuron. The back propagation calculates the gradient of each parameter, and updates all parameters according to some learning rate policy, which is normally called gradient descent. The selection of the learning rate policy has an important impact on the performance of the model.

Apart from the learning rate policy, both parameter initialization and the neural network architecture are important. Parameter initialization decides the initial position for an ML model. As we know, gradient descent means that the ML model will search for a second-best solution on the hyperplane. The initial position at the beginning may lead the model to a relatively poor solution. Gaussian and He are two common initialization methods. The architecture of the neural network contains the numbers of hidden layers and neurons on each layer mainly. More layers and more neurons on each layer will make the DNN's own learning ability powerful, meanwhile, it becomes more difficult to train such a model.

The concepts commonly used in several DNNs are described below:

1) Feature extraction. The purpose of deep learning is to enable the model to automatically extract the potential features of the data and complete the mapping from data to results. The extraction of deep-level valuable features of the data directly affects subsequent learning tasks and is the basis for the entire model training.

2) Multi-layer perceptron. A perceptron is a fully connected neural network structure with one input layer and one output layer through an activation function. Multi-layer perceptrons are fully connected neural networks with hidden layers.

3) Activation function. It is an important component of the artificial neural network (ANN) model. It can help the model to automatically learn and understand complex non-linear functions effectively, and introduce non-linear characteristics into the model.

4) Transfer learning. Transfer learning is to use the trained model and the acquired knowledge to solve the problem in the target domain, and realize the use of a small number of sample sets to train a highly reliable model.

5) Adaptive learning rate. The process of model training is greatly affected by the learning rate. If the set learning rate is large, oscillation will likely occur and the model cannot converge. The model converges slowly, so the adjustment of the learning rate is very important to the training of the model. If the learning rate can change with the training process of the model, the training process of the model and its result are satisfactory.

6) Gradient descent (GD). A greedy algorithm in the training model process, based on the mathematical idea of differentiation, iterates continuously to find the fastest direction of gradient descent.

7) Error back-propagation (BP). Calculate the difference between the estimated value of the model and the actual value, continuously propagate the difference, continuously adjust the model parameters, update the global parameters, and obtain a high reliability model.

8) Regularization. It is used to solve the problem of overfitting of model training, prevent the model from over-extracting data features, and balance the results of model training and the goal of smaller parameter values.

3.2 Fault Location with Convolutional Neural Network

CNN was firstly proposed in Ref. [25], which consists of the convolution layer, the pooling layer, and the fully-connected layer, as Fig. 2(b) shows. The calculation in the convolution layer is sparse, different from that in the fully-connected layer. In the convolution layer, multiple convolution kernels are used to execute a convolutional operation on the input data matrix. Then these values are handled by a nonlinear function to adjust the nonlinear feature. The pooling layer only summarizes a data matrix into a value through a pooling function, for example, max-pooling that outputs the maximal value, min-pooling that outputs the minimal value, avg-pooling that outputs the average value of all elements, etc. The pooling layer aims to reduce the size of data while remaining the most important feature. After several convolution layers and pooling layers, the output will be converted to a one-dimensional array, as the input of the final fully-connected layer. In the fully-connected layer, the extracted features in different granularities and aspects could be mixed and considered together.

CNN is similar to DNN in many aspects. The mechanism of parameter update in CNN is also the combination of forwarding propagation and back propagation. The former is used to calculate the output of each kernel and each neuron for a specific input, the latter is used to calculate the gradients. In weight initialization, learning rate adjustment, and many other configurations, CNN normally takes the same approach as DNN does. Benefited from the extreme reduction of parameter numbers compared with DNN, CNN can take hundreds and thousands of convolution layers to extract more abstract features. However, although CNN has been proved in image classification and some other domains, due to CNN's black-box attribute, it's hard to prove and explain that CNN could be deployed on any domain problem strictly.

The following shows all the elements related to CNN and how they work in detail.

1) Convolution layer. It uses the convolution kernel to

scroll over the input data set according to the stride to extract the characteristics of the input data. The multi-layer convolution layer can better extract the inherent rich features of the data. Convolutional layers have important characteristics that contribute significantly to the performance of the system, such as sparse weights, but convolution operations may cause the loss of information that may exist on image boundaries. Because they are only captured when the filter slides, they may not have a chance to be seen. A simple and effective way to solve this problem is to use zero padding.

2) Pooling layer. The pooling layer is used to merge semantically similar resources in the output of a specific layer of the network. The pooling layer essentially reduces the input by merging adjacent values, so subsequent layers are represented using a smaller network structure. The most common pooling operation is max pooling, which returns the maximum value from a set of zone values.

3) Non-linearity. The next layer after convolution is non-linear. Non-linearity can be used to regulate or cut off the output produced. This layer is applied to saturate the output or limit the output produced. Sigmod, Tanh, and ReLU functions are frequently used non-linear functions. Because the Sigmod and Tanh functions are saturation functions, gradients disappear easily when the network layer is deep, which is suitable for situations where the network layer is not too deep. The gradient of the ReLU function in the positive part is constant, and the gradient of the ReLU function can be zero. It can be used for sparse representation. Therefore, the ReLU function is more widely used when the network level is deeper.

4) Fully connected layer. Each node in the fully connected layer is directly connected to each node in the previous and the next layer. The main disadvantage of the fully connected layer is that it contains many parameters and needs complex calculations in the training example. Therefore, we try to eliminate the number of nodes and connections, and use local connections to meet the requirements of reducing the number of nodes and connections. For example, Lenet and Alexnet are designed as a network of depth and breadth while maintaining computational complexity.

4 Simulation and Results

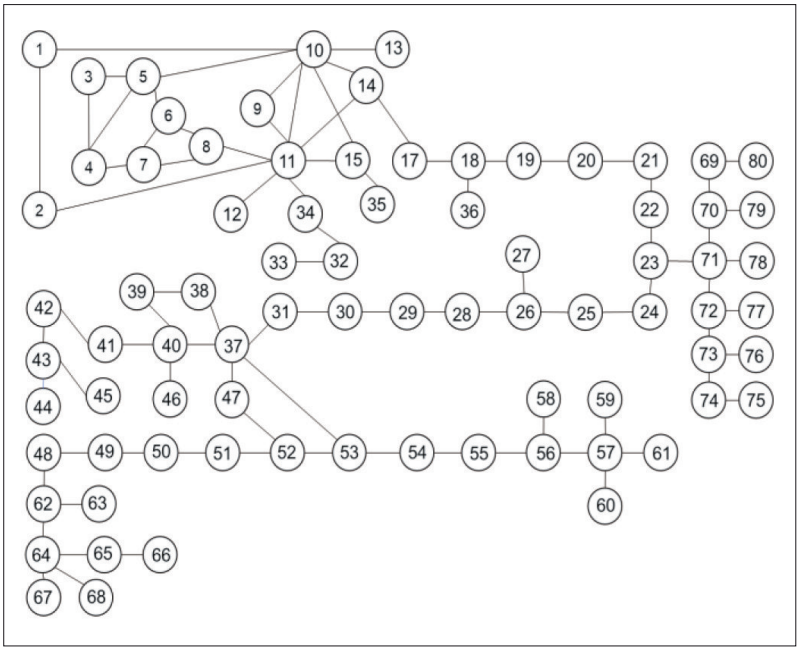
To evaluate the quality of the dataset we built, we introduce some other algorithms as the benchmark. And to compare the performance of CNN and DNN in the fault location, we also execute these algorithms on our dataset. The original data is collected from the commercial optical network as shown in Fig. 3.

4.1 Dataset

Table 1 shows the result of the FP-growth algorithm, which contains six transactions calculated from alarm data. The capital letter means specific alarm type, and the value in parentheses is the support degree. In the sliding window phase, we set w as 10 s, and s as 4 s to get the transaction set, as the second column shows in Table 1. Then the minimal support degree is set as 3, and the remaining transactions after filtering is shown in the third column of Table 1. After the FP-tree is built, we could get the conditional pattern base of the alarm data frequent item as follows.

- A→{}:5
- B→{A}:1, {A, C, D}:1, {C, E}:1
- C→{A}:3, {}:1
- D→{A, C}:3
- E→{A, C, D, E}:2, {C}:1
- F→{A, C, D, E}:2, {A, C, D, B}:1

In this paper, we introduce several classical association rule mining algorithms like MINWAL(w), Apriori, and BEPMWA, to compare the time performance of FP-growth as Figs. 4(a) and 4(b) shows. Also, we add a distributed version of FP-Growth, called FP-Growth-Distributed. Fig. 4(a)

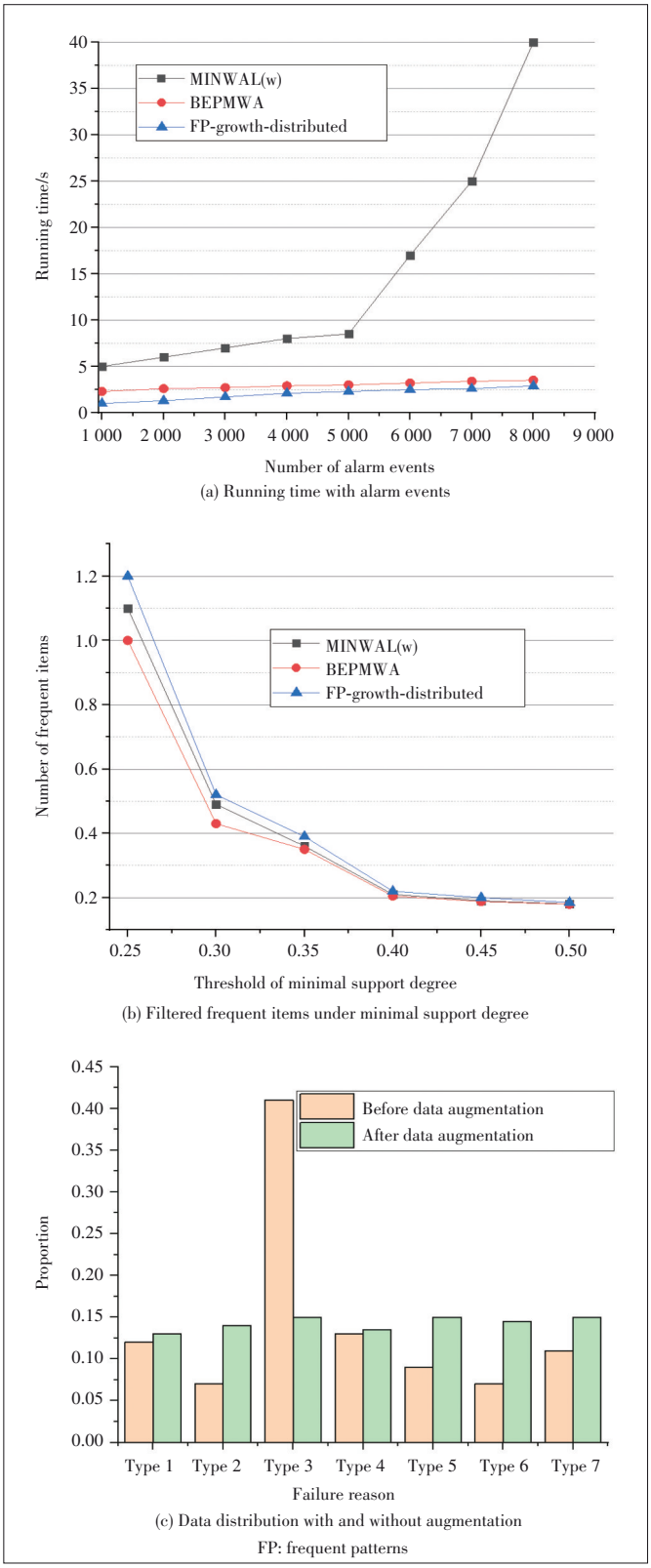


▲Figure 3. Commercial optical network for evaluation

▼Table 1. Frequent item using FP-growth

Transaction ID	Included Alarm Data	Filtered Alarm Data
001	B, A, H, J, P	A(5), B(3)
002	A, D, C, W, U, V, F, E	A(5), C(4), D(3), F(3), E(3)
003	A	A(5)
004	B, C, N, O, E	C(4), B(3)
005	D, B, C, A, Q, F, P	A(5), C(4), B(3), D(3), F(3)
006	D, A, C, G, Q, E, F, M	A(5), C(4), D(3), E(3), F(3)

shows the treatment efficiency for alarm events. The time cost of MINWAL(w) increases rapidly, but the time cost of other algorithms remains a linear growth. MINWAL(w) always consumes the most time, and FP-Growth-Distributed always has the highest efficiency. Fig. 4(b) shows the size of filtered frequent items with different thresholds of the minimal support degree. All algorithms generate about the same size of frequent items when the minimal support degree increases, which means they perform similarly on the frequent item selection. However, when the minimal support degree is small, FP-Growth-Distributed performs better. Then, as mentioned in Section 2, the data augmentation method is executed manually. Fig. 4(c) shows the size comparison before and after data augmentation. Data augmentation balances the distribution of transaction caused by different failure reasons.



▲Figure 4. Results of building an available dataset

4.2 Performance Evaluation

After the process of the previous subsection, an available dataset is built. The input is transaction, and the output is the fault type. In this paper, 8 fault types are introduced, including no fault, fiber interruption, board failure, clock failure, control board failure, power failure, base environment degradation, and line loss. So the size of the output is 8.

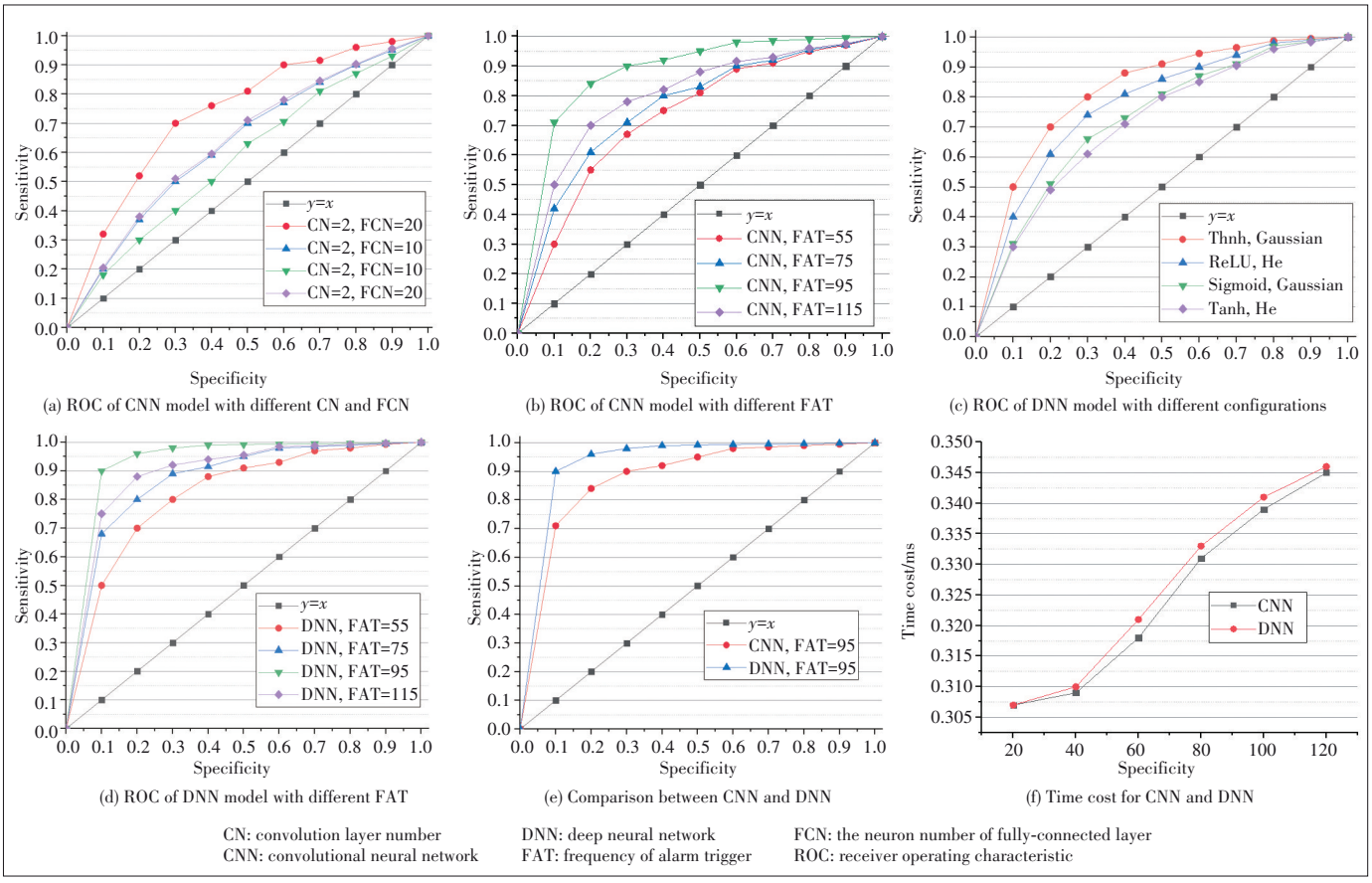
In the evaluation for classification task, TP indicates that network failure happens, and the output of ML algorithm is correct. FP indicates that network failure doesn't happen, but the output makes a mis-judgement. FN indicates that network failure happens, but the output makes a mis-judgement. And TN indicates that network failure doesn't happen, and the output is correct. Then the sensitivity and specificity are described as in Eqs. (1) and (2). We make x-axis as the specificity, and y-axis as the sensitivity to draw receiver operating characteristic (ROC) curves for CNN and DNN in Figs. 5(a) and (b). The area surrounded by line $x=y$ and an arc shows the performance. The performance is better when the area is bigger.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{1}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{2}$$

In Fig. 5(a), CN means the number of convolution layers, and FCN means the neuron number of fully-connected layers. In several different configurations, CNN performs the best when CN is 2 and FCN is 20. Based on such optimal parameter combination, Fig. 5(b) shows the influence of frequency of alarm trigger (FAT). This CNN model reaches its best performance while FAT equals 95.

In Fig. 5(c), we set the learning rate as the adjustable learning rate, and the number of hidden layers is 5. Tanh, ReLU, and Sigmoid are the activation functions in a neuron. Gaussian and He are parameter initialization methods. Ac-



▲Figure 5. Evaluation results for CNN and DNN model

cording to some advanced hyper-parameter filters, we skip many parameter combinations, and only 4 kinds of them remain in Fig. 5(c). In all DNN models, the parameter combination with ALR, Layers=5, Tanh, and Gaussian performs the best. Based on such optimal parameter combination, Fig. 5 (d) shows the influence of FAT on this best DNN model. Same with Fig. 5(b), this DNN model reaches its best performance while FAT equals 95.

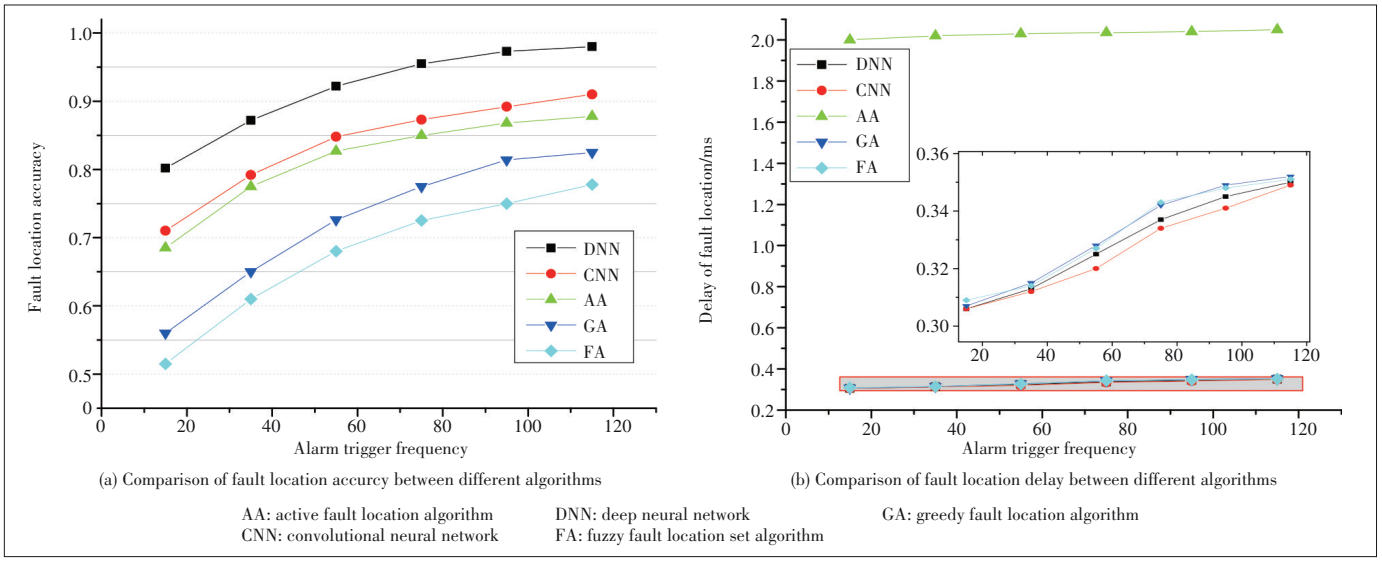
Fig. 5(e) compares the performance of the best CNN model and the best DNN model where FAT equals 95. When the specificity is specified, the sensitivity of DNN is larger than that of CNN, which indicates that the DNN algorithm could perform better than CNN on the fault location dataset. On the other side, in Fig. 5(f), we evaluate the time cost of CNN and DNN under different FATs. DNN costs more time than CNN does, however, the millisecond level is enough for the commercial requirement.

Figs. 6(a) and 6(b) are comparisons of different fault location algorithms in terms of fault location accuracy and location delay. DNN stands for fault location based on deep neural network, CNN represents fault location based on convolutional neural network, AA stands for active fault location algorithm, GA stands for greedy fault location algorithm, and FA is a fuzzy fault location set algorithm[26 - 27]. From Fig. 6 (a), it can be seen that the fault location based on deep learning is inferior to existing methods in terms of location accuracy. Compared with other fault location algorithms, fault location algorithms based on DNN and CNN have higher accuracy. Comparing with the fault location based on DNN and CNN, we find that the former has higher accuracy.

As shown in Fig. 6(b), compared with traditional fault location algorithms, the location delay of fault location based on DNN is more timely. The fault location delay based on DNN is the second only to the fault location based on CNN. The time difference between the DNN and CNN is not much. The specific reason is that the structural level of the fault location model based DNN is deeper than that of CNN, so the fault location speed has a delay of about 0.1 ms.

5 Conclusions

In this paper, based on huge original data collected from commercial equipment, we propose a standard process flow to build a high-quality dataset. Then the supervised machine learning algorithms CNN and DNN are introduced to analyze the fault reason as an auxiliary for fault location. The evaluation shows that when a dataset is built, this method can effectively improve the quality of the dataset, and DNN is more



(a) Comparison of fault location accurcy between different algorithms

(b) Comparison of fault location delay between different algorithms

AA: active fault location algorithm
CNN: convolutional neural network
DNN: deep neural network
FA: fuzzy fault location set algorithm
GA: greedy fault location algorithm

▲**Figure 6. Comparison of fault location accuracy and fault location delay between different algorithms**

suitable than CNN in application.

References

[1] RAFIQUE D, VELASCO L. Machine learning for network automation: overview, architecture, and applications [J]. Journal of optical communications and networking, 2018, 10(10): D126. DOI: 10.1364/jocn.10.00d126
[2] BISHOP C M. Pattern Recognition and Machine Learning [M]. New York, USA: Springer, 2006
[3] TALEBI H A, KHORASANI K, TAFAZOLI S. A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite's attitude control subsystem [J]. IEEE transactions on neural networks, 2009, 20(1): 45 – 60. DOI: 10.1109/TNN.2008.2004373
[4] CARDOSO G, ROLIM J G, ZURN H H. Identifying the primary fault section after contingencies in bulk power systems [J]. IEEE transactions on power delivery, 2008, 23(3): 1335 – 1342. DOI: 10.1109/TPWRD.2008.916743
[5] FANG H Z, SHI H, DONG Y F, et al. Spacecraft power system fault diagnosis based on DNN [C]//Prognostics and System Health Management Conference. IEEE, 2017: 1 – 5. DOI: 10.1109/PHM.2017.8079271
[6] MUSUMECI F, ROTTONDI C, NAG A, et al. An overview on application of machine learning techniques in optical networks [J]. IEEE communications surveys & tutorials, 2019, 21(2): 1383 – 1408. DOI: 10.1109/COMST.2018.2880039
[7] RUIZ M , FRESI F , VELA A P , et al. Service-triggered failure identification/localization through monitoring of multiple parameters [C]//The 42nd European Conference on Optical Communication. ECOC, 2016: 1 – 3
[8] TEMBO S R, VATON S, COURANT J L, et al. A tutorial on the EM algorithm for Bayesian networks: Application to self-diagnosis of GPON-FTTH networks [C]//2016 International Wireless Communications and Mobile Computing Conference. IEEE, 2016: 369 – 376. DOI: 10.1109/IWCMC.2016.7577086
[9] GOSSELIN S, COURANT J L, TEMBO S R, et al. Application of probabilistic modeling and machine learning to the diagnosis of FTTH GPON networks [C]//2017 International Conference on Optical Network Design and Modeling. IEEE, 2017: 1 – 3. DOI: 10.23919/ONDM.2017.7958529
[10] CHRISTODOULOPOULOS K, SAMBO N, VARVARIGOS E. Exploiting network kriging for fault localization [C]//Optical Fiber Communication Conference. OSA, 2016: 1 – 3. DOI: 10.1364/ofc.2016.w1b.5
[11] VELA A P, RUIZ M, FRESI F, et al. BER degradation detection and failure identification in elastic optical networks [J]. Journal of lightwave technology, 2017, 35(21): 4595 – 4604. DOI: 10.1109/JLT.2017.2747223
[12] VELA A P, SHARIATI B, RUIZ M, et al. Soft failure localization during commissioning testing and lightpath operation [J]. IEEE/OSA journal of optical communications and networking, 2018, 10(1): A27 – A36. DOI: 10.1364/JOCN.10.000A27
[13] SHAHKARAMI S, MUSUMECI F, CUGINI F, et al. Machine-learning-based soft-failure detection and identification in optical networks [C]//Optical Fiber Communication Conference. OSA, 2018: 1 – 3. DOI: 10.1364/ofc.2018.m3a.5
[14] RAFIQUE D, SZYRKOWIEC T, AUTENRIETH A, et al. Analytics-driven fault discovery and diagnosis for cognitive root cause analysis [C]//Optical Fiber Communication Conference. OSA, 2018: 1 – 3. DOI: 10.1364/ofc.2018.w4f.6
[15] ABADI M, BARHAM P, CHEN J, et al. TensorFlow: a system for large-scale machine learning [C]//The 12th USENIX Conference on Operating Systems Design and Implementation. USENIX, 2016: 256 – 283
[16] ZHAO Y L, YAN B Y, LIU D M, et al. SOON: self-optimizing optical networks with machine learning [J]. Optics express, 2018, 26(22): 28713. DOI: 10.1364/oe.26.028713
[17] ZHAO Y L, YAN B Y, LI Z T, et al. Coordination between control layer AI and on-board AI in optical transport networks [J]. Journal of optical communications and networking, 2019, 12(1): A49. DOI:10.1364/jocn.12.000a49
[18] YAN B Y, ZHAO Y L, RAHMAN S, et al. Dirty-data-based alarm prediction in self-optimizing large-scale optical networks [J]. Optics express, 2019, 27 (8): 10631. DOI: 10.1364/oe.27.010631
[19] KIM W, CHOI B, HONG E, et al. A taxonomy of dirty data [J]. Data mining & knowledge discovery, 2003, 7(1):81-99. DOI: 10.1023/A:1021564703268
[20] WANG B J, ZHAN Y. A survey and performance evaluation on sliding window for data stream [C]//The 3rd International Conference on Communication Software and Networks. IEEE, 2011: 654 – 657. DOI: 10.1109/ICCSN.2011.6014977
[21] HAN J W, PEI J, YIN Y W. Mining frequent patterns without candidate generation [C]//The 2000 ACM SIGMOD International Conference on Management of Data. ACM, 2000. DOI:10.1145/342009.335372.
[22] BORGELT C. An implementation of the FP-growth algorithm [C]//The 1st International Workshop on Open Source Data Mining Frequent Pattern Mining Implementations. ACM, 2005: DOI: 10.1145/1133905.1133907
[23] HAN J, MORAGA C. The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning [C]//From Natural to Artificial Neural Computation, International Workshop on Artificial Neural Networks. IEEE: 1995.

[24] NAIR V, HINTON G E. Rectified linear units improve restricted boltzmann machines vinod nair [C]//International Conference on International Conference on Machine Learning. IEEE, 2010
[25] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60(6): 84 - 90. DOI: 10.1145/3065386
[26] HUANG Y, SUN S Y, DUAN X S, et al. A study on deep neural networks framework [C]//The IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference. IEEE, 2016: 1519 - 1522. DOI: 10.1109/IMCEC.2016.7867471
[27] LAUZON F Q. An introduction to deep learning [C]//The 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA). IEEE, 2012: 1438 - 1439. DOI: 10.1109/ISSPA.2012.6310529

Biographies

ZHAO Zipiao received the B.S. degree in communication engineering from Hebei Univiversity of Engineering, China in 2017. She is currently a Ph.D. candidate of Beijing Univiversity of Posts and Telecomunications (BUPT), China. Her research focuses on software defined optical networks and machine learning in optiacal networks.

ZHAO Yongli (yonglizhao@bupt.edu.cn) received the B.S. degree in communication engineering and Ph.D. degree in electromagnetic field and microwave technology from Beijing Univiversity of Posts and Telecomunications (BUPT), China in 2005 and 2010, respectively. He is currently a professor of the Institute of Information Photonics and Optical Communications, BUPT. From Jan. 2016 to Jan. 2017, he was a visiting associate professor with UC Davis, USA. He has published more than 300 papers in international journals and conference. Since 2015, he has become a senior member of IEEE. His research focuses on software defined optical networking, elastic optical networks, datacenter networking, and optical network security.

YAN Boyuan received the B.S. degree in communication engineering from Beijing Univiversity of Posts and Telecomunications (BUPT), China in 2015. He is currently pursuing his Ph.D. degree with BUPT. His research interests include software defined optical networking, service function chaining, and network resource allocation with machine learning.

WANG Dajiang received B.S. and M.S. degrees in mechanical and electronic engineering from East China University of Science and Technology , China, and Shanghai University, China in 1996 and 2001 respectively. By far, he has been working with ZTE Corporation for more than 17 years and persistently doing the job related to OTN intelligent management and control. Now he is the Director of Intelligent Optical Network Product Planning in ZTE and mainly responsible for researching autonomous network architecture and solution on OTN. In total, as the chief author, he has issued more than 60 patents in the fields of WASON, SDON, and AN, most of which had been authorized by CNIPA and USPTO and commercialized in ZTE OTN products. Meanwhile, cooperating with other experts in the field, he has also published more than 20 papers in international influential journals and conferences.