A Learning-Based Service Function Chain Early Fault Diagnosis Mechanism Based on In-Band Network Telemetry*

Meiming FU†, Qingyang LIU††, Nonmembers, Jiayi LIU††††, Member, Xiang WANG†††††, and Hongyan YANG†††, Nonmembers

SUMMARY Network virtualization has become a promising paradigm for supporting diverse vertical services in Software Defined Networks (SDNs). Each vertical service is carried by a virtual network (VN), which normally has a chaining structure. In this way, a Service Function Chain (SFC) is composed by an ordered set of virtual network functions (VNFs) to provide tailored network services. Such new programmable flexibilities for future networks also bring new network management challenges: how to collect and analyze network measurement data, and further predict and diagnose the performance of SFCs? This is a fundamental problem for the management of SFCs, because the VNFs could be migrated in case of SFC performance degradation to avoid Service Level Agreement (SLA) violation. Despite the importance of the problem, SFC performance analysis has not attracted much research attention in the literature. In this current paper, enabled by a novel detailed network debugging technology, In-band Network Telemetry (INT), we propose a learning based framework for early SFC fault prediction and diagnosis. Based on the SFC traffic flow measurement data provided by INT, the framework firstly extracts SFC performance features. Then, Long Short-Term Memory (LSTM) networks are utilized to predict the upcoming values for these features in the next time slot. Finally, Support Vector Machine (SVM) is utilized as network fault classifier to predict possible SFC faults. We also discuss the practical utilization relevance of the proposed framework, and conduct a set of network emulations to validate the performance of the proposed framework.

key words: SFC, fault analysis, INT, LSTM, SVM

1. Introduction

Network Function Virtualization (NFV) has emerged as a significant trend for future networks, such as the 5G system and Software Defined Networks (SDN) [1]. Enabled by NFV, network functions such as firewalls and routers are packaged as virtual machines (VMs) running on generic hardware rather than on proprietary hardware. The decoupling of network functions from dedicated hardware provides great flexibility towards network service deployment and management. With NFV, individual virtual network functions (VNFs) are combined together to form vertical network services. For example, in the 5G system, Network Slices are end-to-end logical virtual networks running on top of the 5G physical infrastructure [2]. In data centers, Service Function Chains (SFCs) are typical virtual networks with chaining structure [3]. A SFC is normally composed by an ordered set of VNFs to provide the tailored network services from the underlying physical network.

On one hand, NFV provides programmable flexibilities for the deployment and management of vertical network services. On the other hand, it also brings new management challenges towards future networks [4]. Network service providers should meet tenant’s Service Level Agreements (SLAs) and avoid SLA violations. A SLA defines the level of service expected by a tenant from the network service provider, which is largely affected by the underlying network state, such as end-to-end delay. In this case, SFC performance measurement and analysis is crucial for SFC management. Early SFC fault prediction and diagnosis is beneficial by migrating the corresponding VNFs towards better physical locations to avoid SLA violations. However, performing early SFC fault analysis is challenging due to the fact that it requires to get enough flow-level high quality network measurement data in a manner that is efficient, timely, and flexible.

Network telemetry is a novel generation network measurement technology, which is still in its development [5]. Especially, In-band Network Telemetry (INT) [6] is one of the network telemetry technologies, which provides detailed real-time network states information to the control plane by attaching network state metadata to INT telemetry packets at the line rate in the data plane. INT can be utilized to monitor SDN networks [7], optical networks [8] and industrial wireless sensor networks [9]. In theory, one is able to collect switch-internal information using the INT approach by defining appropriate metadata available from the underlying devices. Besides, INT allows pre-flow and fine-grained real-time traffic monitoring. These characteristics enables INT as a suitable tool for measuring real-time performance of SFCs. Further, SFC’s fault analysis can be conducted based on the collected INT data.

In this work, we design a learning-based framework for SFC’s early fault prediction and diagnosis based on INT collected SFC monitoring data. We first introduce the INT-
based SFC management system architecture, depicted in Fig. 4. Then, we concentrate on analyzing the INT collected SFC performance features data to predict SFC’s early faults. By consulting the state-of-the-art INT switch products, we specifically select a set of network state features, including end-to-end delay, per-link delay, packet ingress/egress timestamps, etc., as the inputs of the framework. Further, a predictor module is utilized to predict the possible values of these features for the next system time slot. We adopt the Long Short-Term Memory (LSTM) networks as the predictor. LSTM is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. Finally, a fault classifier module is used to analyze the state of the SFC based on the predicted features. We apply the Support Vector Machine (SVM) as the fault classifier. SVM is a supervised machine learning algorithm, which is widely utilized in the field of classification. The performance of the proposed framework is evaluated through intensive network emulations. Firstly, we set up a virtual network environment to conduct network emulations based on the Mininet network emulator [10]. We create normal and abnormal traffic flows and obtain the INT telemetry data. This data set is further labeled and utilized to train and test the learning-based framework. Specifically, we optimize the performance of the framework by evaluating multiple different hyperparameters for LSTM and SVM. The evaluation shows that the framework can predict early SFC faults with high accuracy. Finally, we also discuss practical relevance of the framework.

Related work is summarized in Sect. 2. Then, Sect. 3 introduces some necessary background information about INT and SFC. The INT-based SFC early fault analysis method is fully discussed in Sect. 4. In Sect. 5 we evaluate the performance of our proposals. Finally, we conclude this work in Sect. 6.

2. Related Work

The purpose of network measurement serves for network fault diagnosis, which aims to detect network anomalies and locate network failures. Network fault diagnosis has been studied for decades, in which intelligent monitoring system and machine learning mechanism were applied to assist the analysis [11]. With the development of network technology, the concept of SDN and NFV has been proposed. Consequently, virtual network related concepts, such as network slicing and SFCs are widely adopted in future networks, such as the 5G system and cloud data centers. These new techniques provides the network management with efficiency and flexibility. However, since the network becomes more complex, a variety of new network management problems arise. For example, detecting Service Level Agreements (SLAs) violations should be tackled to guarantee the performance of network services carried by virtual networks. In [12], the authors propose a supervised machine learning approach to make use of VNF monitoring data and perform virtual network anomaly detection and root cause localization. By monitoring and analyze resource usage and traffic load of VNFs of virtual network, [13] compared several machine learning algorithms’ performance on detecting SLA violation in multiple scenarios. In [14], the authors apply LSTM to learn and predict the failure of virtual network. However, the above works do not deal with how to obtain the network monitoring data.

Based on programmable data plane technology, INT has emerged as a novel network measurement paradigm and has received extensive attention in both academia and industry in recent years. Many research organizations, institutions are engaged in this domain, such as Internet Engineering Task Force (IETF) and Open Network Foundation (ONF), and many research projects are also founded. Currently, this technology is still in development. INT can be utilized to monitor SDN networks [7], optical networks [8] and industrial wireless sensor networks [9]. There are some works investigate the monitoring of network slices and SDN-based network in the literature. In [15], the authors demonstrate a SDN-based framework that uses INT data for the control and management of slices in IEEE 802.11 networks. However, the analysis of INT monitoring data is neglected. In [16], the authors build a causality model to infer the causal relationship between service related network components and service-level observable symptoms, and utilize INT to monitor these symptoms for locate fault for SDN. In [17], the authors present mechanism to predict and locate network pathes with performance degradation based on INT. With path failures, an alternative forwarding path can be selected to maintain service performance.

There are also some works try to analyze the INT data by utilizing machine learning methods. In [18], the authors apply machine learning classifiers, such as nearest neighbors, naive Bayes, decision tree and SVM, to decide if the target KPI of VNF is normal or abnormal from telemetry data. The target KPI is packet loss, and the collected telemetry data include VNF physical performance data, such as CPU metrics, page fault, memory usage, etc. In [19], the authors propose a network anomaly detection method based on INT and RNN. The RNN model is trained by a collected INT dataset. The dataset is obtained by creating normal and abnormal flows in a virtual network environment. In [20], the authors propose an AI-assisted framework to analyze the INT data for achieving application-aware service provisioning. The network telemetry data is collected from a multilayer INT scheme to monitor an IP-over-EON network. However, the AI-assisted data analysing method is not depicted in detail. In view of the importance of this topic, the combination of network monitoring, INT and machine learning methods needs to be further explored.

INT provides real-time packet-level network telemetry, and the possibility of enhanced network management by analyzing INT telemetry data. However, the introduction of INT also brings monitoring complexity and cost. Firstly, inserting INT metadata into data packets adds overhead and impacts on packet processing time. Moreover, the flexibil-
ity of programmable data plane also brings switch configuration complexity. For example, determining the location and number of INT sinks is non-trivial and largely affects telemetry efficiency. Finally, the INT server treats INT data in a centralized manner which is error-prone and unscalable.

In view of the above disadvantages of INT, in the literature, there are some works try to balance the tradeoff between INT monitoring granularity and cost. In [21], the authors propose a probabilistic INT mechanism to bound the amount of information added to each packet to limit overhead. In [22], the authors study the optimization problem for distributed virtual network monitoring which determines the optimal placement of monitoring nodes and the coordinated forwarding of network monitoring traffic. In [23], the authors present IntOpt, a telemetry system that maps SFC monitoring jobs into a minimum number of monitoring flows for multiple VNF SFC monitoring. In [24], the authors propose In-band Network Function Telemetry that extends the principle of INT to collect VNFs running performance data using INT packets. The above methods are helpful for implementing virtual network’s monitoring system based on INT, and they are complementary to our work.

As a result, the utilization of INT for the monitoring of virtual network is a promising solution for efficient network management. In this work, we propose a mechanism to predict early fault of SFC based on INT telemetry data. The system architecture, which integrates SDN controller, INT telemetry server and slice orchestrator, is presented. Further, INT data analyze process is introduced which applies learning based feature prediction and fault classification.

3. Background

We give some necessary background information on INT and SFC in this section.

3.1 INT

Network measurement is the foundation of network management. As summarized in [7], [25], network measurement can be roughly classified into three categories: (1) the traditional network measurement (including active, passive and hybrid measurement) developed since 1995; (2) the software-defined measurement (including control and data plane measurement) with the development of the software-defined networks (SDN) since 2008; and (3) network telemetry with the rise of programmable data plane (PDP) technology since 2015. A typical PDP technology is Programming Protocol-Independent Packet Processors (P4)[26]. P4 is a high level programming language which specifies how data packets are to be processed inside packet forwarding network entities. In-band Network Telemetry (INT), which is led by P4.org, belongs to the third category. INT is an emerging network telemetry technology that provides fine-grained real-time end-to-end accurate network state monitoring by combining data packet forwarding with network measurement.
serting INT metadata. Theoretically, one can collect multiple types of network information using the INT approach by defining the proper set of metadata that can be provided from the underlying network devices. Typically, INT metadata may include:

1. Switch level: switch ID, L2 and L3 flow table count, timestamp, received packets, received bytes, etc.
2. Port level: ingress and egress port ID, link utilization, bytes received, bytes transmitted, bytes dropped, receive error count, ingress timestamp, egress timestamp, etc.
3. Queue level: queue ID, bytes enqueued, bytes dropped, overrun error count, etc.
4. Packet level: packet’s input port, packet’s output port and packet number count.
5. Flow table and flow level: flow table version, packet lookup count, packet match count, flow count, etc.

By providing accurate per-flow network monitoring data, INT is suitable for upper layer SFC’s performance analysis.

3.2 SFC

Network Service Function Chaining (SFC) is a chain of connected network functions, such as firewalls (FW), network address translation (NAT), traffic load balancer (TLB), deep packet inspection (DPI) etc., to form the required end-to-end network service system provided by network service providers. Enabled by SDN and NFV technologies, SFC can be established in an on-demand fashion and managed with flexibility by inter-connecting a set of VNFs. For instance, a service flow must go through sequentially the firewall, packet inspection and load balancing systems. Then, the service corresponding SFC might be instantiated with a service path: \{FW2 → DPI3 → TLB5\}, where FW2, DPI3 and TLB5 are the selected VNFs. The SFC architecture specifications are addressed by the IETF SFC working group (RFC 7665) and the Open Network Foundation (ONF)\(^2\). SFC is composed of a set of connected VNFs, hence each SFC has its proper traffic flow passing through the ingress point and end with the egress point, known as service function path (SFP) or network forwarding path (NFP). From the viewpoint of physical infrastructure layer, the traffic also follows a path of traffic forwarding entities, such as SDN switches. Hence, the pre-flow network monitoring technique INT is suitable for monitoring the performance of a SFC.

In IETF SFC specification, the SFC data plane contains four components: the SFC classifier, service function (SF), service function forwarder (SFF) and SFC proxy. The SFC classifier locates at the ingress point and differentiates the incoming traffic into flows. The traffic classification mechanism can be header-based or tag-based, such as by using Network Service Header (NSH) or VLAN tag. The SFs are virtualized network functions (VNFs in NFV terminology). The SFFs are traffic forwarding network entities. SFC proxy works for cooperating with SFC-unaware SFs by encapsulating and decapsulating SFC packet headers. Finally, the SFP is the real path (the exact SFFs and SFs) that packets traverse.

Incorporating with SDN and ETSI NFV, the ONF proposed another model for SFC architecture. ONF specification is more focused on the management and orchestration of network services. A network service is carried by VNF forwarding graph (VNFFG) that describes the topology of the virtual network. In this context, VNFs corresponds to SFs. NFP indicates the order of involved VNFs in the VNFFG, which corresponds to SFP. Especially, ONF specifies the SFC SFF as extended SDN switches by supporting SFC packet headers.

Without loss of generality, we consider a unidirectional SFC with acyclic chaining structure. The model of the SFC is given in Fig. 3. Especially, the traffic comes from the ingress point, and passes through a set of network switches and functions, and ends on the egress point. In the model, we consider that each switch is attached with one VNF. For switches with multiple VNFs, the switch can be multiplied on the path. We monitor the data forwarding path: (ingress → \(s_1\) → \(s_2\) ··· \(s_n\) → egress) by utilizing the INT technique.

4. INT-Based SFC Early Fault Analysis

4.1 System Architecture

In this section, we introduce the SFC early fault diagnosis mechanism based on INT. In Fig. 4, we show the architecture of the SFC monitoring and management system. In the control plane, we distinguish three main management entities: the SDN controller, the INT telemetry server and the...
Slice orchestrator.

1. The SDN controller is in charge of managing SDN switches. In the environment of SFC monitor and management, the SDN controller not only manages flow control, but also controls telemetry instructions based on PDP technique. For example, by defining rules for specific VLAN ID and flow ID, the data traffic of a specific SFC can be monitored by inserting INT telemetry information on specific SDN switches. The SDN controller communicates with the other two management entities through East-West interfaces.

2. The INT telemetry server performs two tasks: (1) By consulting the slice orchestrator, INT telemetry server defines telemetry instructions (such as telemetry frequency, flow ID, etc.) and send these information to the SDN controller to inform switches to perform the telemetry; (2) It collects and analyzes telemetry data from switches. We especially detail the INT monitoring data analysis process within Fig. 4. The analysis mainly contains four steps: INT packet data aggregation, network feature extraction, network feature prediction and network fault prediction. We will detail the treatment mechanism in the following sections. By analyzing these data, the network states for future time can be predicted. Further, the state of the SFC can be transmitted towards the Slice orchestrator to manage the SFC. For example, in case of SLA degradation of the SFC, the VNFs and virtual links can be migrated towards other physical network entities to maintain the performance of the SFC.

3. The Slice orchestrator. We still utilize the word slice to incorporate with the concept of network slicing, which is more general than SFC. Typically, SFC is a special kind of network slice with chaining structure. The slice orchestrator is in charge of managing the virtual network by initializing and migrating virtual nodes and links. It specifies the monitoring requirement for each SFC, such as telemetry frequency, telemetry network features, telemetry links, etc., and passes these information to the INT telemetry server to map each SFC to a specific telemetry job.

In the data plane, the SFC is established by initializing corresponding VNFs and connecting them by virtual links. To monitor the performance of the SFC, INT is utilized by inserting telemetry information on each hop of the transmission path. The last hop switch finally reports the INT telemetry information towards the INT telemetry server for further treatment. In this work, we focus on the INT telemetry server. More specific, we focus on the telemetry data processing mechanism that supervise the performance of the SFC and predict early service degradation of the SFC. The mechanism and algorithm proposed in this work can be implemented in the INT telemetry server [28].

4.2 Problem Statement

We denote by \(X_t = \{X_1^t, X_2^t, \cdots, X_N^t\}\) the values of \(N\) network features to be monitored at time slot \(t\). The system collects periodical INT telemetry data and integrate these data to obtain the proper data set for each time slot. Let \(H = \{H_1, H_2, \cdots, H_N\}\) denotes the historical data set for the \(N\) features, where each \(H_i\) = \{\(X_1^{t-3}, X_2^{t-2}, X_3^{t-1}\)\} recodes the values of the feature \(X^i\) for the passing time. Finally, we denote by \(F = \{f_1, f_2, \cdots, f_M\}\) the network fault set. Accordingly, we have to address the following problem: based on the historical feature data set \(H, X^i\), the system predicts the value of the feature for the next time slot \(\hat{X}_1^{t+1}\). Then, based on the predicted values, the system predict the network states for the next time slot by finding a mapping of \(\{\hat{X}_1^{t+1}\} \rightarrow F\). To solve the above problem, in the following, we introduce a learning based framework.

4.3 Overall Process

The overall network telemetry data analyze process is also depicted in Fig. 4, within the INT Server. The process mainly contains four steps: (1) INT data aggregate and extract process; (2) SFC performance features predicting process; (3) SFC fault classification process; (4) and finally the recording database process.

Firstly, INT data are aggregated as in Fig. 5 [29]. INT generates a large amount of telemetry data. The first step is to aggregate these data to obtain one data point on the basis of a user-defined time window period. We utilize the average value to represent the network state for a time window. Then, the SFC performance features are extracted from the aggregated INT telemetry data. In our experiments, we utilized link delay and node processing delay as SFC monitoring features. For link delay, it can be extracted directly from the INT telemetry data. For node processing delay, it can be calculated from packet ingress timestamp and packet egress timestamp. For other types of SFC features, proper INT telemetry metadata should be defined and collected. Then, the obtained features data for the current time slot, together with the historical data sets, is passed to the \(N\) predictors to predict the values of the \(N\) features for the next time slot. Then, the \(N\) predicted SFC features values are passed to the early fault classifier to identify the network state for the next time slot. Finally, based on the real observance, the feature values data is labeled and recorded in the database for further utilization.

In the following, we introduce the mechanism utilized as the predictor and the fault classifier. Especially, we utilize...
LSTM as the predictor and SVM as the fault classifier.

4.4 LSTM-Based Feature Prediction

Recurrent neural networks (RNNs) constitute a class of artificial neural network models containing self-reflective connections for neurons. In addition, hidden units in RNNs also receive a feedback from the previous states to current states. These features make RNN suitable for treat sequential data and time series data. However, RNN has the problem of long-term dependencies. Training RNNs is known to be difficult when time dependencies become long. Then, Long Short Term Memory (LSTM) networks are explicitly designed as a special kind of RNN to avoid the long-term dependency problem. The LSTM network structure differs from the conventional architecture by using the hidden layer as a memory unit, which is also known as cell that contains its state over time. Network measurement data are sequential data, which is suitable to be treated by LSTM.

In Fig. 6, a memory cell contains an input gate, a forget gate, internal state and an output gate is illustrated. Let \( x(t) \) denotes the input value. Let \( h(t - 1) \) and \( h(t) \) denote the output value at time \( t - 1 \) and \( t \). Let \( c(t - 1) \) and \( c(t) \) denote the cell states at time \( t - 1 \) and \( t \). The biases of input gate, forget gate, internal state and output gate are denoted by \( b_i, b_f, b_c \) and \( b_o \). The weights of input gate, forget gate, internal state and output gate are denoted by \( w_i, w_f, w_c \) and \( w_o \), and the recurrent weights of them are \( w_{ir}, w_{rf}, w_{rc} \) and \( w_{ro} \). Finally, let \( a(t) \), \( f(t) \), \( c(t) \) and \( o(t) \) denote the output for input gate, forget gate, internal state and output gate.

Then, the input gate calculates:
\[
a(t) = \sigma (w_i x(t) + w_{ir} h(t - 1) + b_i). \tag{1}
\]
The forget gate calculates:
\[
f(t) = \sigma (w_f x(t) + w_{rf} h(t - 1) + b_f). \tag{2}
\]
The internal state is updated according to the following equation:
\[
c(t) = f(t) \times c(t - 1) + a(t) \times \tanh (w_c x(t) + w_{rc} h(t - 1) + b_c). \tag{3}
\]
The output gate performs regulation on the output of an LSTM cell:
\[
o(t) = \sigma (w_o x(t) + w_{ro} h(t - 1) - b_o). \tag{4}
\]
The final output value is:
\[
h(t) = o(t) \times \tanh (c(t)). \tag{5}
\]
The learning of the LSTM follows the following steps. Firstly, compute the LSTM output using the above equations. Then, compute the error between the result and input data. The error is reversely propagated to input gate, cell and forget gate. Then, the weight of each gate is updated using an optimization algorithm. The above process are repeated until the optimal weights and biases are obtained.

4.5 SVM-Based SFC Fault Classification

SVM has been successfully applied to a number of applications. One of the most common use case is for supervised classification analyses. SVM aims to separate the training data samples by finding an optimal separating hyperplane, that is placed at the maximum distance between the set of support vectors. The support vectors are the sample points that lie at the edge of the class distributions. Suppose the labeled data points are represented by \( \{x_i, y_i\}, i = 1, \ldots , S \), \( y_i \in \{1, -1\} \). The hyperplane is represented by \( w \cdot x + b = 0 \), where \( w \) shows the weight and \( b \) is the bias. Then, the learning task involves the optimization of the following problem:
\[
\min \frac{1}{2} ||w||^2, \quad \text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1, \quad \forall i. \tag{6}
\]
The primal lagrangian of the above problem is:
\[
L = \frac{1}{2} (w \cdot w) - \sum_i \alpha_i (y_i(w \cdot x_i + b) - 1), \tag{6}
\]
where \( \alpha_i \leq 0 \) is the lagrangian multiplier. By substituting the derivatives with respect to \( w \) and \( b \), the Wolfe dual lagrangian is obtained as:
\[
W(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j), \tag{7}
\]
where \( \alpha_i \geq 0 \) and \( \sum_i \alpha_i y_i = 0 \). To allow for nonlinear decision surfaces, the data points can be mapped into a higher-dimensional space through a replacement \( x_i \cdot x_j \rightarrow \phi(x_i) \cdot \phi(x_j) \). Then, the so-called kernel method is applied by using a positive definite kernel function to compute \( K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \). The following kernel functions are mostly utilized:
Linear kernel function:
\[
K(x_i, x_j) = x_i \cdot x_j. \tag{8}
\]
Polynomial kernel function:
\[ K(x_i, x_j) = (x_i \cdot x_j + 1)^d. \]  
\[
\text{Gaussian Radial Basis Function (RBF):}
\]
\[ K(x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right). \]

SVMs were designed for binary classification. The binary SVM can be extended for multiclass by utilizing the one-against-one and one-against-all strategies. For one-against-one, \(M(M-1)/2\) classifiers should be trained, where \(M\) is the number of network faults types. For one-against-all, \(M\) classifiers should be trained. Hence, in this work, we apply the one-against-all multiclass method. Further, the performance and hyperparameter selection of SVMs are discussed in Sect. 5.

5. Experiment

5.1 Network Emulation Setup and DATA Set

We first establish network emulations to obtain the required data set through Mininet\[10\]. Mininet is one of the most well-known network emulators in research and academia for emulating both traditional and software-defined networks. When lack of real network data, network emulation is a practical way for obtaining the required data set. The virtual network environment is made with Mininet + ONOS (SDN controller) + BMv2 (programmable software switch) [7]. Each switch is installed with an INT program implemented with P4 language as an ONOS application program.

Figure 7 shows the SFC topology used in this study. The SFC contains three nodes and two links. We first study a simplest form of SFC, more sophisticated SFCs can be segmented into this simplest form to reduce analyze complexity. In the experiment, Host 1 sends data packets towards Host 3. Hence, Switch 1 is the INT source node, Switch 2 is the INT transit node and Switch 3 is the INT sink node. The Bandwidth of Link 1 is set to 30MBps and that of Link 2 is set to 50MBps. The data rate from Host 1 to Host 3 is set to 10MBps. We consider three kinds of early network faults: Link 1 overload, Link 2 overload and Switch 2 over-  

delay larger than 2.5ms is regarded as overload. We launched simulations and obtained 4910 records. We consider the following hyperparameters that could be considered. In this work, we consider a single hidden layer LSTM, then we especially consider the following hyperparameters:

1. The number of units in the LSTM hidden layer: The number of unit is one parameter of the Keras LSTM implementation. Selecting the optimal number of units for the LSTM hidden layer is necessary. If the number of units is small, LSTM’s prediction performance will deteriorate. However, if the number of units is too large, the model will overfit on the training set.
Table 2  Example of data set

| t  | Link 1 delay | Link 2 delay | Switch 2 delay | end-to-end delay | Link 1 fault | Link 2 fault | Switch 2 fault | Label |
|----|--------------|--------------|----------------|------------------|--------------|--------------|----------------|-------|
| 1  | 14.34        | 9.08         | 5.8            | 32.29            | 0            | 0            | 1              | 1     |
| 2  | 8.71         | 19.15        | 4.64           | 34.94            | 0            | 1            | 1              | 3     |
| 3  | 1.48         | 21.8         | 3.76           | 30.27            | 0            | 1            | 0              | 2     |
| 4  | 32.65        | 2.74         | 4.36           | 41.26            | 1            | 0            | 1              | 5     |
| 5  | 33.36        | 19.29        | 3.2            | 57.19            | 1            | 1            | 0              | 6     |

Table 3  RMSE values for hyperparameters of LSTM for Link 1’s delay

| Units # \ Epoch | 50  | 100 | 150 | 200 | 250 | 300 |
|----------------|-----|-----|-----|-----|-----|-----|
| 10             | 3.64| 3.38| 4.29| 3.96| 3.60| 3.61|
| 20             | 3.60| 3.68| 3.63| 3.60| 3.65| 3.61|
| 30             | 3.67| 3.61| 3.61| 3.61| 3.61| 3.61|
| 40             | 3.62| 3.62| 3.63| 3.61| 3.62| 3.65|
| 50             | 3.62| 3.61| 3.61| 3.63| 3.62| 3.61|

Table 4  MAPE values (%) for hyperparameters of LSTM for Link 1’s delay

| Units # \ Epoch | 50  | 100 | 150 | 200 | 250 | 300 |
|----------------|-----|-----|-----|-----|-----|-----|
| 10             | 9.48| 10.98|10.78|10.08|9.14|9.20|
| 20             | 9.09| 8.85| 8.82| 8.97| 9.59| 9.11|
| 30             | 8.86| 8.88| 8.92| 8.86| 8.89| 8.95|
| 40             | 8.88| 8.84| 9.41| 8.90| 9.34| 9.64|
| 50             | 9.28| 9.23| 8.93| 9.48| 8.88| 9.07|

2. Epoch size: One training epoch is referred to a single iteration over all training instances. Small number of training epochs leads to inaccurate model, whereas large number of training epochs results in model overfitting. Therefore, finding a suitable epoch number is also critical in optimizing the model.

The setting of the hyperparameters is related to the complexity of LSTM training and inference. For training, the setting of epoch determines the number of training iterations, hence larger epoch size implies higher training complexity. The number of units refers to the dimension of hidden state vector, and this parameter affects both training and inference complexity since it affects hidden state matrix multiplication operation. Higher units number also implies higher complexity. We utilize the grid search method for finding the appropriate LSTM model hyperparameters. For each of the four network features, and for each combination of hyperparameters, an LSTM network is designed and trained. Then, the obtained model is evaluated using RMSE and MAPE. We show the RMSE and MAPE values for Link 1’s delay in Table 3 and Table 4. For RMSE the difference between number of units is not significant. For 10 units, the RMSE value is 3.64 for 50 epochs. However, the value increases first as the epoch size increases, and then decreases to 3.60 for epoch size larger than 250. This implies that for 10 units, 50 epochs is not sufficient and longer epoch size results in better performance. For the other number of units, the RMSE values varies slightly with the change of parameters. Then, for MAPE, the value is low for 30 and 40 units with small epoch size. However, the overall trend is not obvious among all these parameters. From the results, several pairs are suitable, for example units number 30 and epoch size 250, and units number 10 and epoch size 250, etc. Considering the complexity mentioned above, we intend to select a model with lower inference complexity, hence we choose units number 10 and epoch size 250 as the selected hyperparameter values. Please note that this selection is not exclusive, other choices are also possible.

Finally, we determined the hyperparameters utilized for the LSTM models for the four network features in Table 5. Especially, for for Link 1’s delay prediction, a single hidden layer LSTM model with 10 neurons and 250 training epochs is utilized. For Link 2’s delay prediction, a single hidden layer LSTM model with 10 neurons and 150 training epochs is utilized. For the switch’s processing delay prediction, a single hidden layer LSTM model with 30 neurons and 200 training epochs is utilized. And for the end-to-end delay prediction, a single hidden layer LSTM model with 40 neurons and 200 training epochs is utilized.

The prediction is finally performed by the above selected LSTM models. We show the prediction in Fig. 8 for Link 1’s delay. The blue line shows the actual value of the data set. 80% of the data set is utilized for training and the rest is utilized for testing. The orange line shows the predicted value of the training set, and the green line shows the predicted value of the testing set. In Table 5, we show the RMSE and MAPE values for the prediction of the four
network features.

5.3 SVM Fault Classification

Based on the predicted SFC features values, SVM is applied to predict the state of the SFC fault. Then, the performance of the prediction is evaluated based on the following metrics: accuracy, precision, recall and F1 score.

\[
\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN}. \tag{13}
\]

\[
\text{precision} = \frac{TP}{TP + FP}. \tag{14}
\]

\[
\text{recall} = \frac{TP}{TP + FN}. \tag{15}
\]

\[
F_1 = \frac{2TP}{2TP + FP + FN}. \tag{16}
\]

The above metrics are typical for binary classification. For multiclass, we further utilize macro- and weighted-values which are defined as follows:

\[
\text{macro}_{\text{acc}} = \frac{1}{F} \sum_{i} \text{accuracy}_i. \tag{17}
\]

\[
\text{macro}_{\text{pre}} = \frac{1}{F} \sum_{i} \text{precision}_i. \tag{18}
\]

\[
\text{macro}_{\text{recal}} = \frac{1}{F} \sum_{i} \text{recall}_i. \tag{19}
\]

\[
\text{macro}_{F_1} = \frac{1}{F} \sum_{i} F_1_i. \tag{20}
\]

\[
\text{weighted}_{\text{acc}} = \frac{1}{F} \sum_{i} w_i \cdot \text{accuracy}_i. \tag{21}
\]

\[
\text{weighted}_{\text{pre}} = \frac{1}{F} \sum_{i} w_i \cdot \text{precision}_i. \tag{22}
\]

\[
\text{weighted}_{\text{recal}} = \frac{1}{F} \sum_{i} w_i \cdot \text{recall}_i. \tag{23}
\]

\[
\text{weighted}_{F_1} = \frac{1}{F} \sum_{i} w_i \cdot F_1_i. \tag{24}
\]

where \(w_i\) shows the percentage of samples of this class in the whole data set.

Then, we determine the hyperparameter for the SVM kernel. We utilized three kinds of kernel functions: linear, polynomial and RBF. For polynomial kernel, the degree \(d\) should be determined. We utilized variety of SVMs with different values of hyperparameters, and evaluate the performance by accuracy, the results are shown in Fig. 9. From the result, \(d\) is set to 4. For RBF, we apply grid searching, and the hyperparameter \(C\) is set to 32768 and \(\gamma\) is set to 0.001248. For more details, the values of the four metrics are further detailed in Table 6. From the result, RBF obtains the best classification performance.

![Fig. 9 Polynomial kernel hyperparameter](image)

| Table 6 SVM fault classification with multiple kernels |
|------------------------------------------------------|
| RBF | polynomial | linear |
| running time (s) | 0.0276 | 0.0676 | 0.9289 |
| macro accuracy | 0.9913 | 0.9740 | 0.9852 |
| precision | 0.9379 | 0.8296 | 0.8777 |
| recall | 0.9226 | 0.7585 | 0.8558 |
| F1 | 0.9293 | 0.7694 | 0.8624 |
| weighted accuracy | 0.9902 | 0.9713 | 0.9871 |
| precision | 0.9661 | 0.9064 | 0.9426 |
| recall | 0.9653 | 0.8958 | 0.9410 |
| F1 | 0.9653 | 0.8937 | 0.9405 |

5.4 Practical Relevance

In this section, we discuss some practical relevance of the proposed method.

1. Firstly, in the emulation, we set up an experiment with a simple SFC topology. For SFC with more sophisticated structure, the number of fault classes increases exponentially, which brings huge complexity to the analysis. In this case, we propose to segment the SFCs into shorter parts, and analyze each part separately, which can greatly reduces analyze complexity.

2. Further, the SFC features can be defined adaptively according to the SFC’s application type. For network related features, INT could provide variety of network monitoring data by properly define the INT metadata fields. For VNF’s features, INT could provide ingress and egress timestamps of data packets, which could reflect the processing rate of the VNF. For more detailed VNF’s physical machine features, such as CPU utilization etc., other probing methods could be applied to obtain these specific data to further facilitate the SFC monitoring and analyzing framework. There are also methods proposed to obtain network function measurement data through INT [24].

3. In the current work, we utilized network emulation and fault injection methods to test our proposed SFC early fault prediction and analysis mechanism. We expect that our proposal could be also applied to real network data. However, the characteristics of real network
data should be also considered. For instance, the treatment of the dataset to obtain the tagged training data set need to be further investigated. Moreover, real network data is biased such that abnormal is rarer than normal data, then the treatment of the dataset to obtain a proper training and testing data also need to be further studied.

4. Finally, the processing framework proposed in Fig. 4 is general. In the current paper, we utilized LSTM as the feature predictor and SVM as the fault classifier. However, other types of predictor and classifier could also be utilized. In the following research work, we seek to compare multiple different types of tools for predictor and classifier.

6. Conclusions

INT has emerged as a novel network measurement paradigm and has received extensive attention in both academia and industry in recent years. The utilization of INT in monitoring SFC’s state is crucial for SFC’s management, and this topic has not been widely discussed in the literature. In this work, we propose an INT-based SFC early fault diagnosis mechanism. Firstly, SFC’s features are extracted from INT monitoring data, then LSTM based feature predictor is applied to predict the features’ value for future time, and SVM based fault classifier is applied to predict the network state based on the predicted feature value. Finally, an experiment is contacted to validate the performance of the mechanism. The utilization of INT measurement data in SFC and network slice performance monitoring is still in the preliminary stage. For future work, we aim to explore multiple different tools for predictor and classifier. Moreover, the treatment of INT monitoring data is also a challenging task in real network environment, we aim to study efficient data processing methods. Further, both supervised and unsupervised learning methods should be explored for inference from network features toward network faults.

References

[1] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “Nfv: state of the art, challenges, and implementation in next generation mobile networks (vepc),” IEEE Netw., vol.28, no.6, pp.18–26, 2014.
[2] C. Mei, J. Liu, J. Li, L. Zhang, and M. Shao, “5g network slices embedding with sharable virtual network functions,” Journal of Communications and Networks, vol.22, no.5, pp.415–427, 2020.
[3] P. Quinn and J. Guichard, “Service function chaining: Creating a service plane via network service headers,” Computer, vol.47, no.11, pp.38–44, 2014.
[4] D.M. Manias and A. Shami, “The need for advanced intelligence in nfv management and orchestration,” IEEE Netw., vol.35, no.1, pp.365–371, 2021.
[5] H. Song, F. Qin, P. Martinez-Julia, L. Ciavaglia, and A. Wang, “Network Telemetry Framework,” Internet-Draft draft-ietf-opsawg-ntf-13, Internet Engineering Task Force, 2021. Work in progress.
[6] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, and L.J. Wobker, “In-band network telemetry via programmable dataplanes,” ACM SIGCOMM Symposium on SDN Research (SOSR), 2015.
[7] V.T. Nguyen, H. Jongsuhan, and H.J. Won-Ki, “Towards onos-based sdn monitoring using in-band network telemetry,” 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp.76–81, 2017.
[8] A. Sgamelluti, A. Giorgetti, F. Paolucci, P. Castoldi, and F. Cugini, “Open source implementation of openconfig telemetry-enabled netconf agent,” 2019 21st International Conference on Transparent Optical Networks (ICTON), pp.1–4, 2019.
[9] A. Karaagac, E. De Poorter, and J. Hoebeke, “In-band network telemetry in industrial wireless sensor networks,” IEEE Transactions on Network and Service Management, vol.17, no.1, pp.517–531, 2020.
[10] B. Lantz, B. Heller, and N. McKeown, “A network in a laptop: Rapid prototyping for software-defined networks,” Proc. 9th ACM SIGCOMM Workshop on Hot Topics in Networks, New York, NY, USA, Association for Computing Machinery, 2010.
[11] C. Hood and C. Ji, “Proactive network fault detection,” Proc. INFOCOM ’97, vol.3, pp.1147–1155, 1997.
[12] C. Sauvanaud, K. Lazri, M. Kaliniche, and K. Kanoun, “Anomaly detection and root cause localization in virtual network functions,” 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pp.196–206, 2016.
[13] J. Hong, S. Park, J.-H. Yoo, and J.W.-K. Hong, “Machine learning based sla-aware vnf anomaly detection for virtual network management,” 2020 16th International Conference on Network and Service Management (CNSM), pp.1–7, 2020.
[14] L. Zhang, X. Zhu, S. Zhao, and D. Xu, “A novel virtual network fault diagnosis method based on long short-term memory neural networks,” 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), pp.1–5, 2017.
[15] P.H. Isolani, J. Haxhibeqiri, I. Moerman, J. Hoebeke, J.M. Marquez-Barja, L.Z. Granville, and S. Latré, “An sdn-based framework for slice orchestration using in-band network telemetry in ieee 802.11,” 2020 6th IEEE Conference on Network Softwarization (NetSoft), pp.344–346, 2020.
[16] Y. Tang, Y. Wu, G. Cheng, and Z. Xu, “Intelligence enabled sdn fault localization via programmable in-band network telemetry,” 2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR), pp.1–6, 2019.
[17] C. Jia, T. Pan, Z. Bian, X. Lin, E. Song, C. Xu, T. Huang, and Y. Liu, “Rapid detection and localization of gray failures in data centers via in-band network telemetry,” NOMS 2020 – 2020 IEEE/IFIP Network Operations and Management Symposium, pp.1–9, 2020.
[18] Z. Zhou and T. Zhang, “Applying machine learning to service assurance in network function virtualization environment,” 2018 First International Conference on Artificial Intelligence for Industries (AI4I), pp.112–115, 2018.
[19] S. Nam, J. Lim, J.-H. Yoo, and J.W.-K. Hong, “Network anomaly detection based on in-band network telemetry with rnn,” 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), pp.1–4, 2020.
[20] J. Kong, B. Niu, S. Tang, Y. Li, H. Fang, W. Lu, and Z. Zhu, “Network nervous system: When multilayer telemetry meets ai-assisted service provisioning: (invited paper),” 2019 18th International Conference on Optical Communications and Networks (ICOCN), pp.1–3, 2019.
[21] R. Ben Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, “Pint: Probabilistic in-band network telemetry,” Proc. Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM ’20, New York, NY, USA, pp.662–680, Association for Computing Machinery, 2020.
[22] R. Hohemberger, A.F. Lorenzon, F. Rossi, and M.C. Luizelli, “Optimizing distributed network monitoring for nfv service chains,” IEEE Commun. Lett., vol.23, no.8, pp.1332–1336, 2019.
[23] D. Bhamare, A. Kassler, J. Vestin, M.A. Khoshkhoghi, and J. Taheri, “Intopt: In-band network telemetry optimization for nfv
service chain monitoring,” ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pp.1–7, 2019.

[24] J. Liang, J. Bi, Y. Zhou, and C. Zhang, “In-band network function telemetry,” Proc. ACM SIGCOMM 2018 Conference on Posters and Demos, SIGCOMM 2018, Budapest, Hungary, Aug. 20-25, 2018, pp.42–44, ACM, 2018.

[25] L. Tan, W. Su, W. Zhang, J. Lv, Z. Zhang, J. Miao, X. Liu, and N. Li, “In-band network telemetry: A survey,” Computer Networks, vol.186, p.107763, 2021.

[26] P. Bosshart, D. Daly, M. Izzard, N. McKeown, J. Rexford, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, “Programming protocol-independent packet processors,” ACM SIGCOMM Computer Communication Review, vol.44, pp.87–95, 12 2014.

[27] A.M. Medhat, T. Taleh, A. Elmangoush, G.A. Carella, S. Covaci, and T. Magedanz, “Service function chaining in next generation networks: State of the art and research challenges,” IEEE Commun. Mag., vol.55, no.2, pp.216–223, 2017.

[28] N.V. Tu, J. Hyun, G.Y. Kim, J.H. Yoo, and J.W.K. Hong, “Intcollection: A high-performance collector for in-band network telemetry,” 2018 14th International Conference on Network and Service Management (CNSM), pp.10–18, 2018.

[29] A. Pellegrini, P.D. Sanzo, and D.R. Avresky, “A machine learning-based framework for building application failure prediction models,” 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, pp.1072–1081, 2015.

[30] H. Abbasimehr, M. Shabani, and M. Yousefi, “An optimized model using lstm network for demand forecasting,” Computers & Industrial Engineering, vol.143, p.106435, 2020.

Meiming Fu received his Bachelor’s degree from Qingdao Binhai University. He is Engineer of Power Line Communication Application Technology Laboratory of State Grid. His research focuses on power line communication, 4G communication, 5G communication.

Qingyang Liu received his Masters degree from Huazhong University of Science and Technology. He is senior engineer of Power Line Communication Application Technology Laboratory of State Grid. His research focuses on power line communication, 4G communication, 5G communication.

Jiayi Liu obtained her Bachelors of Science degree in Electronic Engineering from Xidian University in Xian China, in 2007. She received her Master of Science and Ph.D., both in Computer Science, from Telecom-Bretagne and Rennes 1 University, France, in 2009 and 2013, respectively. Since 2014, she works as Lecturer with Xidian University, in School of Telecommunication Engineering and Guangzhou Institute of Technology. Her research interests include 5G network, content distribution, Network resource scheduling.

Xiang Wang received his Masters degree from Beijing Institute of Technology (BIT). He is Director of Power Line Communication Application Technology Laboratory of State Grid. His research focuses on chip technology, power information and communication technology.

Hongyan Yang received the B.S. degree in Xidian University, Xian, China, in 2018, where she is currently studying for a masters degree in school of communication engineering of Xidian University. Her main research direction is 5G network slicing.