International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

IJIEMR Transactions, online available on 2nd Jan 2021. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-09&issue=ISSUE-12

DOI: 10.48047/IJIEMR/V09/I12/154

Title: **SELF-LEARNING AND EFFICIENT HEALTH-STATUS ANALYSIS FOR A CORE ROUTER SYSTEM**

Volume 09, Issue 12, Pages: 898-904

Paper Authors

BAKI MAHESHWARI, SADAM MOUNIKA, D. APOORVA, K. SONY

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

SELF-LEARNING AND EFFICIENT HEALTH-STATUS ANALYSIS FOR A CORE ROUTER SYSTEM

BAKI MAHESHWARI[1], SADAM MOUNIKA[2], D. APOORVA[3], K. SONY[4]

[1,2,3,] B TECH Students, Department of CSE, Princeton Institute of Engineering & Technology For Women, Hyderabad, Telangana, India.

[4] Assistant Professor, Department of CSE, Princeton Institute of Engineering & Technology For Women, Hyderabad, Telangana, India.

Abstract: The health status of core router systems needs to be analyzed efficiently in order to ensure high reliability and timely error recovery. Although a large amount operational data is collected from core routers, due to high computational complexity and expensive labor cost, only a small part of this data is labeled by experts. The lack of labels is an impediment towards the adoption of supervised learning. We present an iterative selflearning procedure for assessing the health status of a core router. This procedure first computes a representative feature matrix to capture different characteristics of time-series data. Not only statistical-modeling-based features are computed from three general categories, but also a recurrent neural network-based autoencoder is utilized to capture a wider range of hidden patterns. Moreover, both minimum-redundancy-maximum-relevance (mRMR) method and fully-connected feedforward autoencoder are applied to further reduce dimensionality of extracted feature matrix. Hierarchical clustering is then utilized to infer labels for the unlabeled dataset. Finally, a classifier is built and iteratively updated using both labeled and unlabeled dataset. Field data collected from a set of commercial core routers are used to experimentally validate the proposed health-status analyzer. The experimental results show that the proposed feature-based selflearning health analyzer achieves higher precision and recall than the traditional supervised health analyzer as well the currently deployed rule-based health analyzer. Moreover, it achieves better performance than the three anomaly detection baseline methods under the transformed binary classification scenario.

Index Terms: Feature extraction and selection, auto-encoder, self-learning health-status analysis, time-series analysis, machinelearning techniques, and core router systems.

I. INTRODUCTION

The core network, also referred to as the network backbone, is responsible for the transfer of a large amount of traffic in a reliable and timely manner. The network devices (such as routers) used in the core network are complex hardware/software systems that are vulnerable to hard-todetect/hard-to-recover errors [1]. Consider a multi-card chassis system, which is a widely used architecture in core routers. A range of failures can occur in such a complex system:

• Hardware failures: A multi-card chassis system can have tens of separate cards, and each card can have hundreds of components. Since each component consists of hundreds of advanced chips, each chip in turn has hundreds of I/Os and millions of logic gates, and the operating frequency of chips and I/Os are now in the GHz range [2] [3], the number of incorrect hardware behaviors can be very high.

Moreover, in such a complex system, whenever a hardware failure occurs, it is difficult for debug technicians to accurately identify the root cause of this failure and take effective corrective actions [4] [5] [6]. • Software failures: Since the throughput of modern multicard chassis system is approaching Tbps levels, failures caused by subtle interactions between parallel applications have become more frequent [5] [7]. Traditional reactive fault tolerance aims at repair after failures occur [7]. However, this approach needs to spend a significant amount of time to identify and repair faults, which can stall system operation. System specifications require nonstop utilization (i.e., 99.999% uptime) of core routers deployed in the network backbone [1]. Proactive fault tolerance is more promising because it takes preventive action before a failure occurs [7]. The state of the system is monitored in a real-time manner. When system degradation is determined via health assessment, proactive repair actions such as job migration can be executed to avoid errors, thereby ensuring the non-stop system utilization [8] [9] [10]. The effectiveness of proactive fault-tolerance solutions depends on whether the health status of core routers can be accurately identified in a timely manner [11] [12]. However, little research has focused thus far on analyzing the long-term health status in a high-performance communication system. Therefore, in this paper, we present the design of an efficient self-learning health analyzer that can be applied to a commercial core router system. We evaluate this method using field data collected from a set of commercial core routers.

II. RELATED REVIEW

A common way to identify a system's health status is to feed its features to an anomaly detector to see whether any data points are statistical outliers. Anomaly detection has been widely used in domains such as intrusion detection and fraud detection [14] [15]. Three types of techniques have been studied in the literature to detect anomalies in time-series data [14]. Unsupervised distance-based anomaly detection utilizes a distance measure between a pair of time-series instances to represent the similarity between these two timeseries. Window-based anomaly detection divides time-series instances into overlapping windows. Anomaly scores are first calculated per window, and then aggregated for comparison with a predefined threshold. In supervised prediction-based anomaly detection, a machine-learning-based predictive model is first learned from historical logs. Next, predicted values are obtained by feeding test data to this predictive model. The predicted values are then compared with the actual measured data points. The accumulated difference between these predicted and the actual observations is defined as the anomaly score for each test time-series instance. Recently, a hybrid anomaly detector has been proposed in [16] to overcome the drawback that a single class of anomaly detection methods is effective for only specific types of time-series. However, a time-series-based anomaly detector is not adequate to obtain the health status of monitored core routers. First, an anomaly detector can only provide information about the anomalous points; patterns before or after anomalies are not revealed, which may also be necessary for predicting failures. Second, an anomaly detector can provide little useful information if no anomalies are identified. However, learning different normal patterns is also important because it can reveal how healthy a core router system is and how different task

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

scenarios can affect the system. Therefore, a health-status analyzer is needed for core router systems. For example, when an anomaly detector triggers an alarm for overheating of boards, it neither reveals root causes nor gives any advance warning about the consequences of this event. In contrast, a health-status analyzer not only tracks how a system gradually entered this overheating state, but also predicts how the system will be affected by this anomalous behavior. The design of a health-status analyzer is more difficult than the implementation of an anomaly detector because: (1) Anomaly detection is unsupervised while health analyzer requires fully-labeled data. However, the volume of operational data collected from commercial core routers can reach TB levels, making it infeasible for experts to label the data manually; (2) Classifying complex time-series data is harder than detecting anomalous time-series data because subtle differences between a pair of time series must also be identified by the classifier. Although a symbol-based health analyzer for core routers was recently proposed [17], it still requires fully-labeled data during its training phase. Data instances are considered as being labeled only after the expert team has determined their normal/abnormal conditions, which is difficult to obtain in the early stages of monitoring. Moreover, although symbolization can reduce the time cost as well as the storage requirement, some critical local information may be lost during symbolization. Therefore, in this work, we use feature extraction and selection techniques as well as a deep-learning-based autoencoder to characterize complex time series. A self-learning approach is then implemented to analyze the health status of core routers using partially labeled data.

III. Methodology

Feature-Based Self-Learning Health Analysis: The key idea in the proposed method (Fig. 2) is that instead of directly analyzing the health status from a large volume of raw time series data, we first extract and select a set of features that capture the characteristics of high-dimensional time series. The notation of a feature in this paper is different from the definition of a feature in previous work, where features refer to the temporal measurements of different monitored items (variables) in core routers. The "features" in this work are defined as metrics calculated from the raw time series of the variables, and they represent various local and global characteristics of the time series. The steps involved in our procedure are as follows:

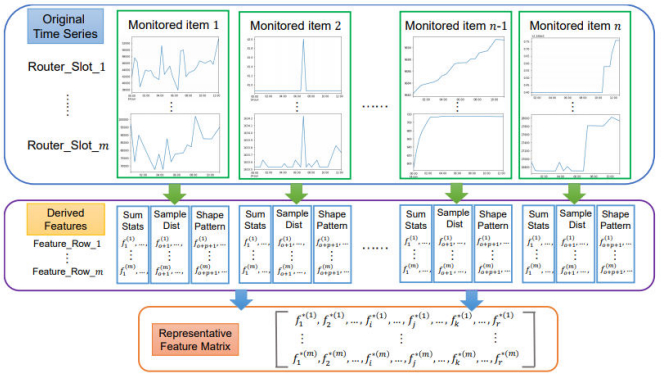

Figure 1: Illustration of feature extraction and selection

Feature extraction and selection: Since each feature is a low-dimensional measurable characteristic of the time series, extracting and selecting a set of representative features provides a more complete understanding of the time series. This component takes a set of clean and aligned time series as input, and outputs a representative feature matrix to the self-learning component. (2) Expert identification: This component is maintained and updated by an expert team. The experts first label a limited

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

number of time series instances using historical warning logs and their rule tables. This set of labels then serves as the initial label vector to the self-learning component. During the self-learning procedure, newly updated labels are also fed to this component for checking. (3) Self-learning component: This component consists of two parts—clustering and classification. The objective of clustering is to increase the number of labeled instances. Since similar instances are grouped together after clustering, the label value of labeled instances can be propagated to unlabeled instances within the same cluster. The classification part is used to identify the health status of the system by iteratively learning a model from partially labeled data.

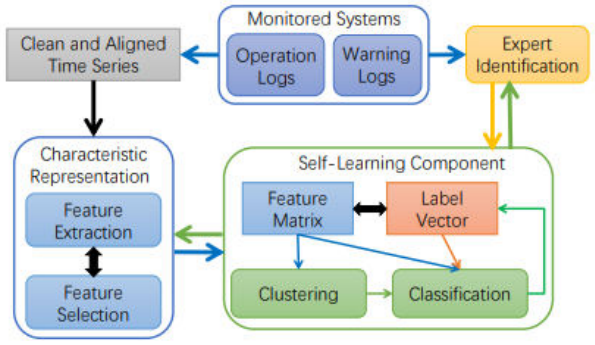

Figure 2: An illustration of the proposed feature-based and self-learning health status analyzer.

Feature Extraction and Selection: Assume that m router slots are monitored, where each router slot has n monitored items across the temporal domain. Therefore, a total of m × n time-series instances are collected in the original dataset: n D = d (1) 1 , d(1) 2 , ..., d(1) n , d(2) 1 , ..., d(2) n , ..., d(m) 1 , ..., d(m) n o , where d (i) j represents the time series sequence extracted from the jth monitored item in the router slot i: d (i) j = {t1, t2, ..., tv}, where v is the number of time points in d (i) j . A set of feature metrics F = {F1, F2, ..., Fu} is then computed using d (i) j to capture various characteristics of this time-series sequence. Specifically, three types of feature metrics [19] are considered:

Auto-encoder-based Feature Learning: Feature extraction and selection step described above suffers from two limitations. First, we have observed from our experiment that even after the mRMR-based feature selection, the number of features is still much larger than the number of available instances, making it difficult for some types of classifiers to be effective. Moreover, some of the previously extracted features are sparse, making it possible to further compress the feature matrix without significant information loss. Second, such a feature extraction step is ad hoc and depends on the experience of experts. Although various characteristics of time series have been extracted, it is hard to ascertain whether the extracted features are sufficient to cover most characteristics of the time-series data. It is possible that some critical characteristics are missed. Therefore, in this paper, the LSTM-based auto-encoder is utilized to capture a wider range of hidden patterns. These new approaches are more general and they better match realistic scenarios.

The traditional artificial neural network (ANN) is a supervised machine learning method that is widely used for pattern classification and related problems. The autoencoder is an unsupervised variant of ANN for learning an efficient representation (encoding) of a set of data. As shown in Figure 5, the simplest form of an autoencoder is a threelayer feedforward artificial neural network. It consists of an input layer, an output layer and a hidden layer. Neurons are

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

arranged in layers, and weighted connections link the neurons in different layers. An autoencoder network can be generally divided into two parts: the encoder that compresses the data from input layer into a short code and the decoder that uncompresses that code and produces a reconstruction in the output layer. Therefore, the objective of an autoencoder is to learn a reduced but meaningful representation that can reconstruct the original data as much as possible. The behavior of an autoencoder depends on both the weights (synaptic strength of neuron connections) and the transfer function (input-output function of neurons).

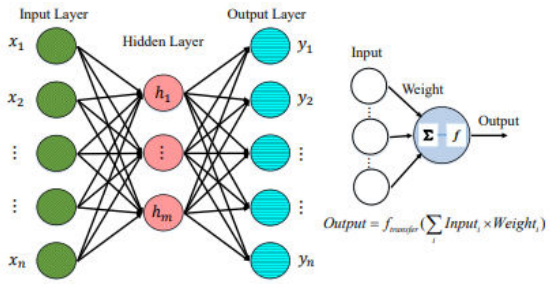

Figure 3: An example of a three-layer autoencoder

Traditional fully-connected feed-forward neural network are not suitable for encoding/decoding time-series data in an autoencoder because of the non-stationary dynamics/patterns within the temporal ordering of the input. Instead, the recurrent neural network (RNN) is promising because it maintains an internal state of the network via a directed cycle of connection between neurons, which allows it to exhibit dynamic temporal behavior [25]. In addition, the hidden state in RNNs is shared over time and thus can contain information from an arbitrarily long window. The Long Short Term Memory (LSTM) serves as the RNN architecture used in autoencoder because it explicitly introduces a memory unit, called the cell, into the network so that long-term historical information can be recalled as needed [26]. As shown in Figure 6, the original feed-forward neural network encoder and decoder are now replaced by multiple LSTM cells. In this framework, the input time-series data are first fed to LSTM encoder cells step-by-step. An encoded representation is then learned from hidden states or outputs of these LSTM cells. This compressed representation is then fed as inputs to the LSTM decoder cells, generating the output time series that closely matches the original input data

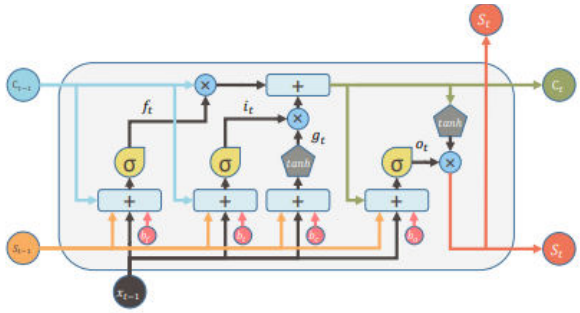

Figure 4: Illustration of the Long Short Term Memory (LSTM) method

Self-learning for Health Analysis: After the feature matrix has been obtained from the original high-dimensional time-series data, it is fed to the self-learning component to train a model for health-status analysis. Since the input data are partially labeled, the learned model and the labeled set are updated iteratively, as shown in Figure 8. Step 1: Initially, the input data consist of the labeled set L = L0 and the unlabeled set U = U0. The percentage of unlabeled data is then calculated: rU = |U| |L|+|U| . If rU is larger than a predefined threshold α, the amount of labeled data is insufficient and a clustering procedure (Step 2)

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

is needed to enrich the labeled set. Otherwise, the input data are directly fed to the classifier learning component (Step 3) Step 2: Clustering is performed to propagate labels between similar instances. Specifically, hierarchical agglomerative clustering (HAC) with link constraints [27] is applied to the input data (LUU), generating a set of clusters C = {c1, c2, ..., ch}. Each cluster ci can contain both labeled and unlabeled instances. The label value of a labeled instance in ci is then propagated to its neighboring unlabeled instances in ci ; Uci is used to denote such a set of unlabeled instances that are now labeled by cluster ci . A set of unlabeled instances is formed: UC = Uc1 ∪Uc2 ∪...∪Uch . The original labeled and unlabeled sets are thus updated accordingly: L = L∪UC , U = U −UC .

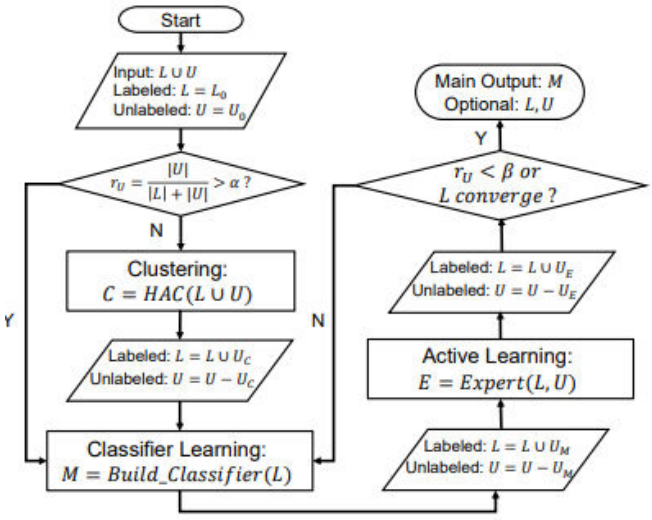

Figure 5: The computation flow of the self-learning component

EXPERIMENTS AND RESULTS

We carried out experiments using the "NE40E" core router product. The details of this core router are shown in Figure 10. It consists of a number of different functional units, such as the main processing unit (MPU), line processing units (LPUs), switch fabric units (SFUs), etc. Also, different types of interface and protocols are supported. A total of 40 core routers were monitored in real time in the field by a distributed agent-based system. A total of 450 multivariate time-series instances were collected over 60 days of operation. Each timeseries instance has 10 monitored items (variables) and 2880 time points. The feature extraction component is then applied to the collected data to extract a wide range of characteristics. The information regarding features extracted from univariate time series is shown in Table I. A total of 623 features are extracted for each univariate time-series instance. Since each instance in 450 time series has 10 variables, a 450 × 6230 raw feature matrix is formed after feature extraction.

Number of Slots	22 slots: 2 MPUs (1:1 backup), 4 SFUs (3+1 backup), 16 LPUs
Environment	0°C to 45°C
Power Consumption	9040W (480G)
Interface type	100GE/40GE, GE/FE, …
Software Version	V8
Supported Protocols	IPv4, IPv6, MPLS, …

Figure 5: Description of the commercial core router used in our experiments.

Without any loss of generality, six labels were defined in our work to represent the overall health status of experimental core routers: (1) Class 0: the system is running in a healthy manner without any obvious abnormal operations (8 out of 450 instances); (2) Class 1: the system is running normally with some minor suspect characteristics (7 instances); (3) Class 2: the system is in relatively good condition with some anomalies (10 instances); (4) Class 3: the system is in a suspect unhealthy warning state (8

instances); (5) Class 4: the system's performance and efficiency are severely affected by critical faulty components (11 instances); (6) Class 5: the system is encountering severe health problems that prevent it from continuing most normal operations (13 instances). The remaining 393 instances are all unlabeled data. An example of these six health labels used in our experiments is shown in Figure 5. We can see that the number of matched abnormal patterns increases while the number of matched normal patterns decreases from "Health Level 0" to "Health Level 5". Note that with the improvement of experts' experience, a larger number of categorical labels or even continuous metric values can be used in the future to define the overall system health status in a more comprehensive way.

IV. CONCLUSION

We have presented a feature-based self-learning health analyzer for a complex core router system. First, both the statistical-modeling-based feature extraction and auto-encoder based feature learning have been utilized to capture different characteristics of time-series data. Next, in the self-learning framework, the model for health analysis is iteratively updated using both labeled and unlabeled data. The effectiveness of the health analyzer has been validated using a comprehensive set of field data collected from a set of commercial core routers.

REFERENCES

[1] M. Medard and S. S. Lumetta, "Network reliability and fault tolerance," ´ Encyclopedia of Telecommunications, 2003.

[2] F. Ye et al., "Board-level functional fault diagnosis using artificial neural networks, support-vector machines, and weighted-majority voting," IEEE Trans. CAD, vol. 32, pp. 723–736, 2013.

[3] Z. Zhong, G. Li, Q. Yang, J. Qian, and K. Chakrabarty, "Broadcast-based minimization of the overall access time for the ieee 1687 network," in VLSI Test Symposium (VTS), 2018 IEEE 36th, pp. 1–6, 2018.

[4] S. Tanwir et al., "Information-theoretic and statistical methods of failure log selection for improved diagnosis," in ITC, 2015.

[5] B. Schroeder et al., "A large-scale study of failures in high-performance computing systems," in Proc. DSN, pp. 249–258, 2006.

[6] S. Jin, F. Ye, Z. Zhang, K. Chakrabarty, and X. Gu, "Efficient board-level functional fault diagnosis with missing syndromes," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, pp. 985–998, 2016.

[7] P. A. Lee and T. Anderson, Fault Tolerance: Principles and Practice, vol. 3. Springer Science & Business Media, 2012.

[8] C. Wang et al., "Proactive process-level live migration in hpc environments," in Proc. Supercomputing, pp. 43:1–43:12, 2008.

[9] S. Jin, Z. Zhang, K. Chakrabarty, and X. Gu, "Toward predictive fault tolerance in a core-router system: Anomaly detection using correlationbased time-series analysis," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, pp. 2111–2124, 2018.

[10] S. Jin and K. Chakrabarty, "Data-driven resiliency solutions for boards and systems," in 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), pp. 244–249, 2018