Anomaly-Detection-Based Failure Prediction in a Core Router System

Shi Jin
and Krishnendu Chakrabarty

Duke University
Durham, NC, USA
Email: shi.jin@duke.edu
krish@ee.duke.edu

Zhaobo Zhang,
Gang Chen and Xinli Gu

Huawei Technologies Co. Ltd.
San Jose, CA, USA
Email: zhaobo.sc.zhang@huawei.com
Gang.C@huawei.com, xinli.gu@huawei.com

***Abstract*—Prognostic health management is desirable for commercial core router systems to ensure high reliability and rapid error recovery. The effectiveness of prognostic health management depends on whether failures can be accurately predicted with sufficient lead time. However, directly predicting failures from a large amount of historical logs is difficult. We describe the design of an anomaly-detection-based failure prediction approach that first detects anomalies from collected time-series data, and then utilizes these "outliers" to predict system failures. A feature-categorizing-based hybrid anomaly detection is developed to identify a wide range of anomalies. Furthermore, an anomaly analyzer is implemented to remove irrelevant and redundant anomalies. Finally, a Support-Vector-Machine (SVM)-based failure predictor is developed to predict both categories and lead time of system failures from collected anomalies. Synthetic data generated using a small amount of real data for a commercial telecom system, are used to validate the proposed anomaly detector and failure predictor. The experimental results show that both our anomaly detector and failure predictor achieve better performance than traditional methods.**

Keywords—Anomaly Detection; Failure Prediction; SVM.

I. Introduction

A core router is responsible for the transfer of a large amount of traffic in a reliable and timely manner in the network backbone [1]. The complex hardware and software architectures of core router systems make them more vulnerable to hard-to-detect/hard-to-recover errors [2]. For example, a wide range of failures can occur in a complex multi-card chassis core router system:

- Hardware failures: The cards that constitute the chassis system and the components that constitute a card can encounter hardware failures. Moreover, connectors between cards and interconnects between different components inside a card are also subject to hard faults. A multi-card chassis system can have tens of different cards, each card can have hundreds of components, and 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. Such high complexity and operating speed lead to an increase in incorrect or inconsistent hardware behaviors. Moreover, in such a large-scale complex system, whenever a hardware failure occurs in the chassis system, it is difficult for debug technicians to accurately identify the root cause of this failure and take effective corrective action [3][4].
- Software failures: The entire chassis system and each card have their own software platforms to control and manage different network tasks. However, since the performance requirement of network devices in the core layer is approaching Tbps levels, failures caused by subtle interactions between parallel threads or applications have become more frequent. These failures often arise because software applications tend to distribute their tasks into parallel agents in order to improve performance [4][5].

All these different types of faults can cause a core router to become incapacitated, necessitating the design and implementation of fault-tolerant mechanisms for reliable computing in the core layer.

Due to the non-stop utilization (99.999% uptime) requirement of core router systems deployed in the network backbone, a traditional fault-diagnosis system is of limited applicability here because it aims at repair after failures occur. Such solutions inevitably stall system operation. In contrast, prognostic health management is promising because it monitors the system in real time, triggers alarms when anomalies are detected, and takes preventive actions before a system failure occurs. Therefore, it ensures non-stop utilization of the entire system [6]. The effectiveness of prognostic health management depends on whether system failures can be predicted in a timely manner [7]. Therefore, in this paper, we present the design of an efficient anomaly-detection-based failure predictor that can be applied to a commercial core router system.

The remainder of this paper is organized as follows. Section II discusses the anomaly detection and failure prediction problems in more detail and highlights the contributions of this paper. In Section III, a number of time-series-based anomaly detection techniques are discussed and a feature-categorization-based hybrid method is proposed. Then, a correlation-based anomaly analyzer is described to select representative anomalies. Section IV discusses how to predict failures based on historical anomaly events. In Section V, experimental results on a synthetic data set generated from a commercial core router system are used to demonstrate the effectiveness of the proposed method. Finally, Section VI concludes the paper and discusses future works.

II. Problem Statement

Prognostic health management can benefit from reasoning-based and data-driven techniques [8], as shown in Fig. 1. The system is monitored by recording different Key Performance Indicators (KPIs). The logged KPI data is then fed to an anomaly detector. When anomalies occur, an anomaly analyzer can be used to filter redundant and irrelevant anomalies. Then, a failure predictor can be triggered to forecast the occurrences of different system failures [7]. Finally, appropriate preventive actions can be executed on the monitored system to avoid failures in advance. We can see that the anomaly detector and failure predictor are two essential components in a data-driven prognostic health management scheme.

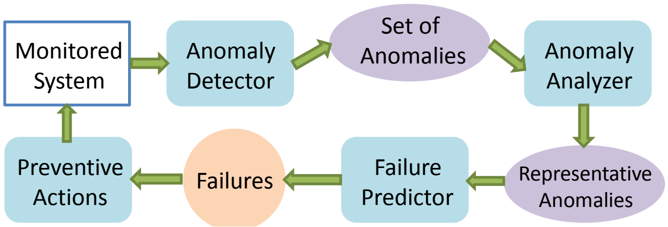


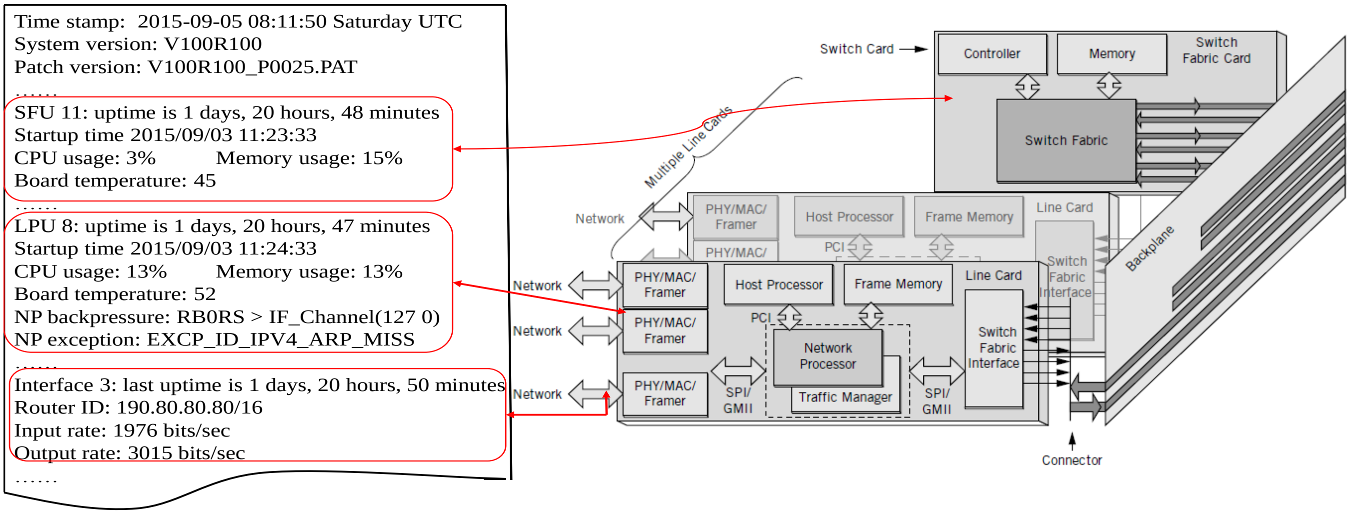
Figure 1. An illustration of data-driven prognostic health management.

Anomaly detection, which is also sometimes referred to as outlier detection, has been widely used in other domains, e.g., intrusion detection and fraud detection [9][10]. For example, density-based techniques, such as K-Nearest Neighbor (KNN) have been used in detecting outliers in high-dimensional datasets [11]. Machine-learning methods, such as Artificial Neural Networks (ANN) have also been applied to detect fraud in large multivariate databases [12]. A Multivariate State Estimation Technique (MSET) has been used to reduce or eliminate No-Trouble-Found [13]. This technique is sensitive to subtle changes in the signal trend, making it effective for detecting indirect anomalies [14].

Failure prediction has also been studied to assess the reliability, availability and serviceability of complex computing systems [15]. For example, a semi-markov reward model has been used to forecast the resource exhaustion problem in software systems [16]. Machine-learning methods, such as Naive Bayes have also been applied to predict hard disk drive failures [17]. A rule-based model has been built to predict attacks in computer networks and illegal financial transactions [18]. However, little research has focused thus far on combining failure prediction with anomaly detection in a high-performance and complex communication system.

The difficulty of developing an efficient anomaly detector and failure predictor for a complex communication system can be attributed to the reason that features extracted from communication systems are far more complex than those from a general computing system. For example, as shown in Fig. 2, a multi-card chassis core router system uses monitors to log a large amount of features from different functional units. These features include performance metrics (e.g., events, bandwidth, throughput, latency, jitter, error rate), resource usage (e.g., CPU, memory, pool, thread, queue length), low-level hardware information (e.g., voltage, temperature, interrupts), configuration status of different network devices, and so on. Each of these features can have significantly different statistical characteristics, making it difficult for a single type of anomaly-detection/failure-prediction technique to be effective.

We, therefore, address the important practical problem of designing an anomaly-detection-based failure predictor that can be effectively applied to a commercial core router system. To achieve this, a feature-categorization-based hybrid method is developed to detect a wide range of anomalies; a correlation-based anomaly analyzer is implemented to select the most important anomalies; and a machine-learning-based failure predictor is developed to predict different failures from



```
Time stamp:  2015-09-05 08:11:50 Saturday UTC
System version: V100R100
Patch version: V100R100_P0025.PAT
......
SFU 11: uptime is 1 days, 20 hours, 48 minutes
Startup time 2015/09/03 11:23:33
CPU usage: 3%         Memory usage: 15%
Board temperature: 45
......
LPU 8: uptime is 1 days, 20 hours, 47 minutes
Startup time 2015/09/03 11:24:33
CPU usage: 13%        Memory usage: 13%
Board temperature: 52
NP backpressure: RB0RS > IF_Channel(127 0)
NP exception: EXCP_ID_IPV4_ARP_MISS
......
Interface 3: last uptime is 1 days, 20 hours, 50 minutes
Router ID: 190.80.80.80/16
Input rate: 1976 bits/sec
Output rate: 3015 bits/sec
......
```

Figure 2. A multi-card chassis core router system and a snapshot of extracted (monitored) features.

historical anomalies.

III. Anomaly Detection and Analysis

In complex communication systems, such as a core router, data is collected in the form of time-series. Three kinds of techniques have been studied in the literature to detect anomalies in time-series data [9]. The first one is distance-based anomaly detection, which utilizes a distance measure between a pair of time-series instances to represent the similarity between these two time-series. The smaller the overall "distance" is, the closer this pair of time-series instances would be. Instances far away from others will be identified as being abnormal. The second one is window-based anomaly detection. This method divides time series instances into overlapping windows. Anomaly scores are first calculated per window and then aggregated to be compared with a predefined threshold. Only when the overall anomaly score of a single time-series instance significantly exceeds a predefined threshold, this instance will be identified as being abnormal. The third one is prediction-based anomaly detection. First, a machine-learning-based predictive model is learned from historical logs. Next, predicted values are obtained by feeding test data to this predictive model. These 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.

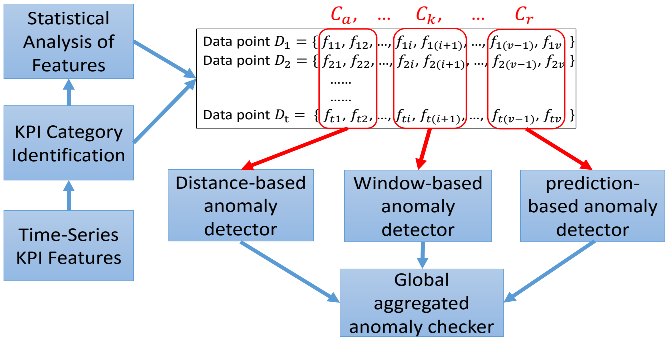

Statistical Analysis of Features
KPI Category Identification
Time-Series KPI Features
C_a, … C_k, … C_r
Data point $D_1 = \{f_{11}, f_{12}, \ldots, f_{1i}, f_{1(i+1)}, \ldots, f_{1(v-1)}, f_{1v}\}$
Data point $D_2 = \{f_{21}, f_{22}, \ldots, f_{2i}, f_{2(i+1)}, \ldots, f_{2(v-1)}, f_{2v}\}$
……
……
Data point $D_t = \{f_{t1}, f_{t2}, \ldots, f_{ti}, f_{t(i+1)}, \ldots, f_{t(v-1)}, f_{tv}\}$
Distance-based anomaly detector
Window-based anomaly detector
prediction-based anomaly detector
Global aggregated anomaly checker

Figure 3. A depiction of feature-categorization-based hybrid anomaly detection.

However, a single class of anomaly detection methods is effective for only a limited number of time-series types. Therefore, we propose a feature-categorization-based hybrid method whereby each class of features can be classified by the most appropriate anomaly detection method. Fig. 3 illustrates the proposed feature-categorization-based hybrid anomaly detection. First, time-series data of different features extracted from the core router system is fed to a KPI-category identification component. Since features belonging to different KPI categories often exhibit significantly different statistical characteristics across the timeline, natural language processing techniques are utilized here to ensure that different KPI categories, such as configuration, traffic, resource type, and hardware can be identified. However, it is also possible that features belonging to different KPI categories have similar trend or distribution across time intervals; therefore, a statistical analysis component is needed to ensure that all features that exhibit similar statistical characteristics are placed in the same class. After these steps, a data point D_t with v features can be divided into different groups $C_a, C_b, \ldots, C_k, \ldots, C_r$, where each group has different statistical characteristics. Next, each group of features is fed to the anomaly detector that is most suitable for this type of features. Finally, the results provided by different anomaly detectors are aggregated so that we can detect an anomaly in terms of the entire feature space.

Although the proposed feature-categorization-based hybrid method can help us detect a wide range of anomalies, not all anomalies are useful and necessary for predicting system failures. First, the temporal and spatial localities of neighboring components lead to co-occurrences of similar anomalies. Second, some anomalies are caused by workload variations or temporary external noise, which makes them irrelevant for predicting system failures. Since the number of possible anomalies will increase from hundreds to tens of thousands when more new features are identified and extracted from the raw log data, anomaly analysis is needed in order to remove irrelevant and redundant anomalies before predicting failures.

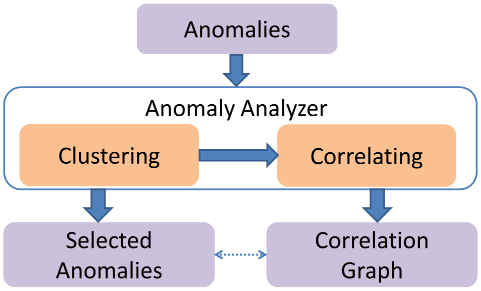


Figure 4. Overview architecture of the proposed Anomaly Analyzer.

Fig. 4 presents an outline of the proposed anomaly analyzer. A set of detected anomalies is fed to the anomaly analyzer. It then goes through two components: the clustering component and the correlating component in sequential order. The clustering component groups anomalies that occur ”simultaneously” (within the same small time interval) and have similar statistical characteristics. Only one anomaly is selected from each cluster and then fed to the next component. The correlating component first identifies both linear and non-linear relationships among these anomalies and then group anomalies that have strong correlations. Finally, the anomaly analyzer outputs a number of correlated anomaly groups. An effective anomaly subset can be generated by selecting the most representative anomalies from these correlated groups. Furthermore, detailed relationships among anomalies within each group can be represented by a correlation graph $G = (V, E)$, where the set of vertices V represent anomalies and the set of edges E represent correlations between anomalies. Therefore, a correlation graph is generated for each group of anomalies.

IV. Failure Prediction

Fig. 5 shows the temporal relationship between faults, anomalies, and failures. Assume that a fault occurs in the system at time point t_r. After a period of time, a wide range of anomalies begin to appear at time point t_{as}. Finally, at time point t_f, the system encounters a fatal failure and crashes. Two important time intervals are defined here: δt_l, referred as the lead time, is the time interval between the occurrence of the last anomaly and the occurrence of the predicted failure. It is defined as $\delta t_l = t_f - t_{ae}$. Only if this lead time is larger than the time required to take preventive actions can our prediction become useful in reality. The parameter δt_d is defined as the time interval between the occurrence of the first and last anomaly, i.e., $\delta t_d = t_{ae} - t_{as}$. Since our failure prediction is based on the detected anomalies in the system, δt_d can be considered to the temporal length of our dataset.

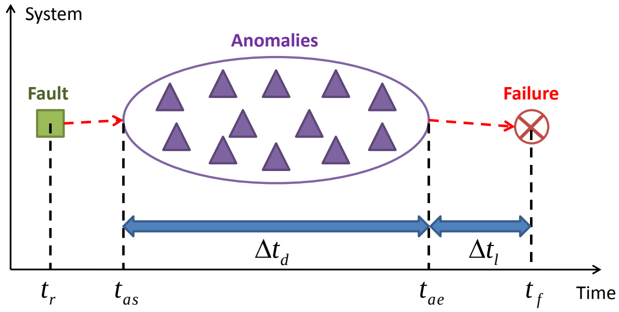


Figure 5. Temporal relationship between faults, anomalies, and failures.

Using the proposed anomaly detector and analyzer, representative anomalies can be identified and recorded. Correlating them with logs of system failures, two types of anomaly event set can be formed: failure-related anomaly event set and non-failure-related anomaly event set. An example of these two types of anomaly event sets is shown in Fig. 6. We can see that A_i represents the ID of each anomaly and F_j represents the ID of each failure. The failure-related anomaly event set consists of records that always end with a failure event F_j while the non-failure-related anomaly event set consists of records that do not have any failure events.

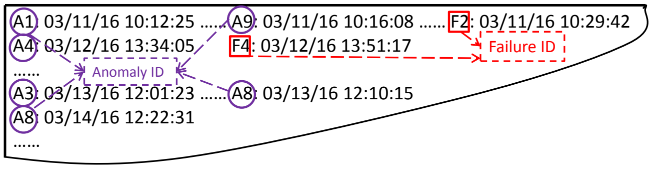

```
A1: 03/11/16 10:12:25 ......A9: 03/11/16 10:16:08 ...... F2: 03/11/16 10:29:42
A4: 03/12/16 13:34:05        F4: 03/12/16 13:51:17
......
A3: 03/13/16 12:01:23 ......A8: 03/13/16 12:10:15
A8: 03/14/16 12:22:31
......
```

Figure 6. An example of anomaly event sets.

An efficient failure predictor should not only predict whether failures will occur, but also predict the type/category and occurrence time of those failures. Therefore, as shown in Fig. 7, the proposed failure predictor consists of two main components: the classifier and the regressor so that both the category and the lead time of failures can be predicted. First, the historical logs including both failure-related and non-failure-related anomaly events are fed as training data to both the classifier and the regressor in order to build corresponding learning models. Second, a set of newly detected anomalies of is fed to these learning models. Finally, the learnt classifier outputs which type of system failures will be triggered by the current anomalies, and the learnt regressor will output the predicted lead time for this type of system failure.

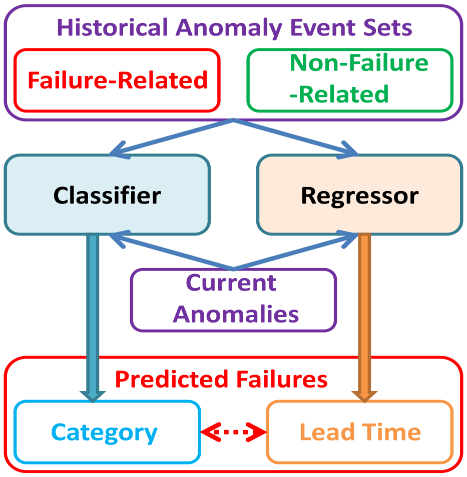


Figure 7. Overview architecture of the proposed failure predictor.

One key step implicit in Fig. 7 is to build training datasets from historical anomaly event sets for both the classification component and the regression component. Suppose we have identified a set of anomalies $\mathbf{A} = \{A_1, A_2, ..., A_N\}$ and a set of system failures $\mathbf{F} = \{F_1, F_2, ..., F_M\}$ from our historical log $\mathbf{H}$. The training dataset $\mathbf{D}$ for the classification component can then be built. For each record H_i in the historical log, it can contain one or more anomalies and either no failure or one failure. If the anomaly A_j appears in the record H_i, $D_{ij} = 1$, otherwise $D_{ij} = 0$. Note that $D_{i(N+1)}$ represents the failure category of the record H_i: If the failure F_k appears in the record H_i, $D_{i(N+1)} = k$. If no failures occur in the record, $D_{i(N+1)} = 0$. The process of building the training dataset $\mathbf{T}$ for the regression component is similar. The only difference is that the occurrence times of anomalies and failures needs to be included now. If the anomaly A_j appears at time t_j in the record H_i, $T_{ij} = t_j$, otherwise $T_{ij} = 0$. If the failure F_k appears at time t_k in the record H_i, $T_{i(N+1)} = t_k$. If no failures occur in the record, $T_{i(N+1)} = 0$.

Different machine-learning techniques can be applied for classification and regression in the proposed failure predictor. Among these techniques, the Support Vector Machine (SVM) algorithm offers several advantages, such as overfitting control through regularisation parameters and performance improvement via custom kernels [19]. Therefore, we utilize SVM-based techniques in this paper. Specifically, we apply multiclass SVM for the classification component and support vector regression for the regression component.

V. Experiments and Results

The commercial core router system used in our experiments consists of a number of different functional units, such as the main processing unit, line processing unit, switch fabric unit, etc. A total of 602 features are monitored and sampled every 5 minutes for 15 days of operation of the core router system, generating a set of multivariate time-series data consisting of 4320 time points.

To evaluate the performance of the proposed anomaly detection and failure prediction methods, we use a 4-fold *cross-validation* method, which randomly partitions the extracted time series dataset into four groups. Each group is regarded as a test case while all the other cases are used for training. The Success Ratio (SR), referred to as a percentage, is the ratio of the number of correctly detected anomalies/predicted failures to the total number of anomalies/failures in the testing set. For example, a SR of 70% means that 7 out of 10 anomalies are correctly detected. In addition to SR, the Non-False-Alarm Ratio (NFAR) is also considered as an evaluation metric. It is defined as the ratio of the number of correctly detected anomalies/predicted failures to the total number of alarms flagged by the anomaly detector/failure predictor.

A. Anomaly Detection and Analysis

To evaluate the effectiveness of feature-categorization-based hybrid anomaly detection, five base algorithms are implemented: KNN is a distance-based anomaly detection method, and for each test instance, its distance to its kth nearest neighboring instance will be considered as its anomaly score. Window-based KNN and window-based SVM are two window-based methods, and the difference between them is the way they calculate their per-window anomaly score. SVR and AR are two prediction-based methods, and the difference is that the former one uses support vector regression to predict values while the latter uses auto-regression for forecasting.

The results are shown in Fig. 8-9. We can see that for the six anomaly detection methods, i.e., KNN, Window-based KNN, window-based SVM, SVR, AR, and the feature-categorization-based hybrid method, the success ratios are 82.7%, 84.5%, 86.4%, 88.2%, 78.6% and 95.1%, respectively, and the non-false-alarm ratios are 73.1%, 76.3%, 80.7%, 88.1%, 71.6% and 92.1%. The reason that the proposed feature-categorization-based hybrid method achieves much higher SR and NFAR than other methods is that it can overcome the difficulty of adopting a single class of anomaly detection to features with significantly different statistical characteristics.

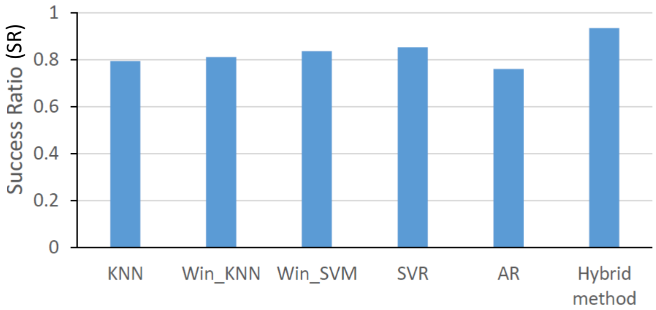


Figure 8. Success ratios of different anomaly detection methods.

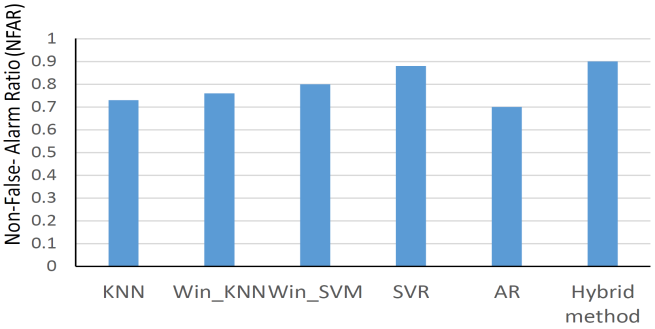


Figure 9. Non-False-Alarm ratios of different anomaly detection methods.

Initially, 467 anomalies are detected by the proposed anomaly detector. The anomaly analyzer can then partition these anomalies into disjoint clusters based on their inner-similarity and inter-correlation. The results of such clustering and correlating are summarized in Table I. We can see that only 15 out of 467 anomalies are identified as being in independent groups (clusters with a single element), which implies that most anomalies are correlated. Moreover, if we choose a single anomaly within each cluster to represent this cluster, only 105 anomalies are needed to represent the entire anomaly set, reducing the number of anomaly dimensions by 77.5%.

TABLE I. RESULTS AFTER CLUSTERING AND CORRELATING OF ANOMALIES.

Size of clusters	Number of clusters	Number of anomalies
1	15	15
2	38	76
3	14	42
4	9	36
6	10	60
10	13	130
15	4	60
21	1	21
27	1	27

B. Failure Prediction

To evaluate the effectiveness of the SVM-based classifier and the SVR-based regressor in the proposed failure predictor, two base algorithms are implemented. For the classification component, a rule-based approach is used: first, a rule model is built from the historical log; each rule takes the form "IF $\{$anomaly A_1, anomaly A_2, ..., anomaly $A_i\}$, THEN $\{$failure $F_j\}$". Second, for each new anomaly set, if a matched rule can be found, the failure label of that rule is assigned to the new anomaly set; otherwise, a random failure label is assigned. For the regressor component, a simple linear regression method is used to predict the lead time of a failure from the occurrence time of its related anomalies.

Fig. 10-11 show the SR and NFAR values for the SVM-based and the rule-based approaches. Eight failure categories are identified from the historical log, and are denoted as A, B, ..., H in the figures. The results can be summarized as follows:

1) For all eight failure categories, the SVM method achieves higher SR and NFAR than the rule-based method. One

possible explanation is that the effectiveness of the rule-based method highly depends on whether the rule model covers a sufficient range of “IF anomalies, THEN failure” rules. However, there are always new anomaly sets that do not match any existing rules, and therefore cannot be predicted well by the rule-based method. In contrast, the SVM method can learn “implicit rules” during its training phase, making it more suitable for predicting failure categories of new anomaly patterns.

2) Both methods perform better in predicting failure categories A and G, but they are worse in predicting failure categories C and F. After analyzing the anomaly event sets related to these failure categories, we find that the anomaly event sets for A and G are significantly different while the anomaly event sets for C and F are very similar. In some cases, the anomaly sets of C and F have exactly the same anomaly events and the only difference is the sequential order of these anomaly events. Since both SVM and rule-based methods do not take this sequential information into consideration, it is quite likely that misclassification will occur when predicting the failure category C and F.

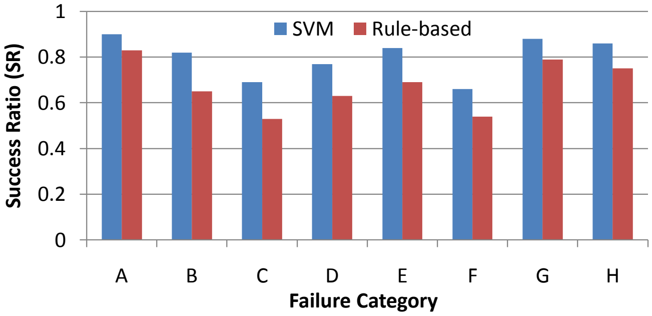


Figure 10. Success ratios of two failure category prediction methods.

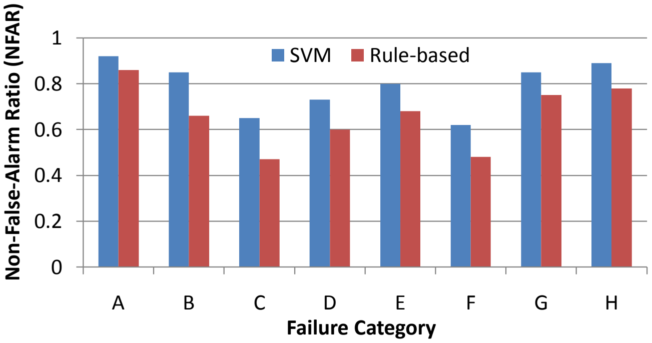


Figure 11. Non-False-Alarm ratios of two failure category prediction methods.

The classical metric Root Mean Square Error (RMSE) is used to evaluate the effectiveness of the failure lead time prediction methods. For our experiments, we define RMSE as the square root of the average squared distance between the actual lead time and the predicted lead time. The results are shown in Fig. 12. We can see that the SVR method achieves much lower RMSE than the linear regression method.

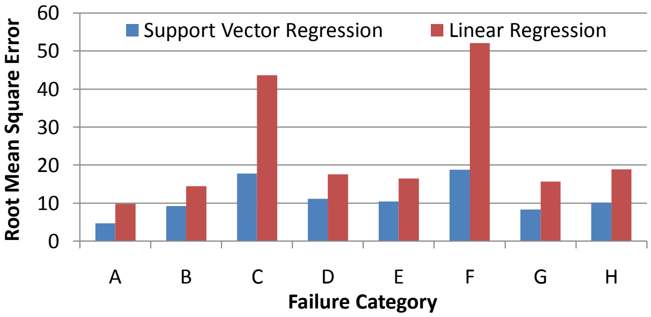


Figure 12. RMSE of two failure lead time prediction methods.

The reason is that in most cases, the temporal relationships between anomalies and failures are not linear. Also, we find that the RMSE for the SVR method for most failure categories is not greater than 10 minutes, which means the lead time predicted by the SVR method can be considered as a realistic approximation.

VI. Conclusion and Future Works

We have described the design of a anomaly-detection-based failure predictor for a complex core router system. Both a feature-categorization-based hybrid anomaly detector and a correlation-based anomaly analyzer have been implemented to detect and identify important anomalies. A SVM-based failure predictor has also been developed to predict the category and lead time of different failures from anomaly event sets. Data collected from a commercial core router system has been used to evaluate the effectiveness of the proposed methods. The experimental results show that the proposed anomaly-detection-based failure predictor achieves not only higher success ratio and non-false-alarm ratio than traditional rule-based method in predicting failure categories, but also lower root mean square error than linear regression method in predicting failure lead time.

However, several drawbacks exist in current work and need to be addressed in the future:

1) The proposed anomaly detector did not take correlations among features into account. Therefore, this method cannot capture anomalies caused by abnormal combination of multiple features. A correlation engine will be investigated in the future to detect multivariate anomalies.
2) The proposed failure predictor did not take sequential information of anomaly events into consideration. Therefore, this method cannot accurately identify failure categories if they share similar anomaly event set. A time-series-based failure predictor will be investigated in the future to better forecast different types of failures.
3) A key assumption in current work is that failures and anomalies are well-correlated. However, this assumption may not always hold true in real scenarios. Whether a sequence of anomalies will trigger a specific failure depends on a variety of factors such as software aging, hardware update, or even human intervention. Therefore, data from other sources such as business scenarios, system configurations, expertise experiences will be incorporated

ISBN: 978-1-61208-500-5

and investigated in the future to build more fine-grained relationships among anomalies and failures.

Acknowledgment

The work of S. Jin and K. Chakrabarty was supported in part by a grant from Huawei Technologies Co. Ltd.

References

[1] V. Antonenko and R. Smelyanskiy, “Global network modelling based on mininet approach.” in Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 145–146.

[2] M. Médard and S. S. Lumetta, “Network reliability and fault tolerance,” Encyclopedia of Telecommunications, 2003.

[3] S. Tanwir, S. Prabhu, M. Hsiao, and L. Lingappan, “Information-theoretic and statistical methods of failure log selection for improved diagnosis,” in Proc. IEEE International Test Conference (ITC), 2015, pp. 1–10.

[4] B. Schroeder and G. Gibson, “A large-scale study of failures in high-performance computing systems,” in Proceedings of the International Conference on Dependable Systems and Networks, 2006, pp. 249–258.

[5] P. K. Patra, H. Singh, and G. Singh, “Fault tolerance techniques and comparative implementation in cloud computing,” International Journal of Computer Applications, vol. 64, 2013, pp. 1–6.

[6] C. Wang, F. Mueller, C. Engelmann, and S. L. Scott, “Proactive process-level live migration in HPC environments,” in Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, 2008, pp. 43:1–43:12.

[7] A. Gainaru, F. Cappello, M. Snir, and W. Kramer, “Fault prediction under the microscope: A closer look into HPC systems,” in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, 2012, pp. 77:1–77:11.

[8] H. H. Chen, R. Hsu, P. Yang, and J. J. Shyr, “Predicting system-level test and in-field customer failures using data mining,” in Proc. IEEE International Test Conference (ITC), 2013, pp. 1–10.

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Computing Surveys (CSUR), vol. 41, 2008, pp. 15:1–15:58.

[10] A. Patcha and J.-M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” Computer Networks, vol. 51, 2007, pp. 3448–3470.

[11] Y. Liao and V. R. Vemuri, “Use of k-nearest neighbor classifier for intrusion detection,” Computers & Security, vol. 21, 2002, pp. 439–448.

[12] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion detection using neural networks and support vector machines,” in Proc. Int. Joint Conf. Neural Networks, vol. 2, 2002, pp. 1702–1707.

[13] F. Bockhorst, K. Gross, J. Herzog, and S. Wegerich, “MSET modeling of crystal river-3 venturi flow meters,” in Proc. Int. Conf. Nuclear Engineering, 1998, pp. 1–17.

[14] K. Vaidyanathan and K. Gross, “MSET performance optimization for detection of software aging,” in Proc. IEEE Int. Symposium on Software Reliability Engineering (ISSRE), 2003.

[15] F. Salfner, M. Lenk, and M. Malek, “A survey of online failure prediction methods,” ACM Comput. Surv., vol. 42, 2010, pp. 10:1–10:42.

[16] K. Vaidyanathan and K. S. Trivedi, “A measurement-based model for estimation of resource exhaustion in operational software systems,” in Proc. International Symposium on Software Reliability Engineering, 1999, pp. 84–93.

[17] G. Hamerly and C. Elkan, “Bayesian approaches to failure prediction for disk drives,” in Proc. International Conference on Machine Learning, 2001, pp. 202–209.

[18] R. Vilalta, C. V. Apte, J. L. Hellerstein, S. Ma, and S. M. Weiss, “Predictive algorithms in the management of computer systems,” IBM Syst. J., vol. 41, 2002, pp. 461–474.

[19] C. Cortes and V. Vapnik, “Support-vector networks,” Journal of Machine Learning, vol. 20, 1995, pp. 273–297.