Outage Prediction and Diagnosis for Cloud Service Systems

Yujun Chen[1,2,*], Xian Yang[2], Qingwei Lin[2], Honyu Zhang[3], Feng Gao[4], Zhangwei Xu[4], Yingnong Dang[4], Domgmei Zhang[2], Hang Dong[2], Yong Xu[2], Hao Li[2], Yu Kang[2]
[1]Beihang University, Beijing, China
[2]Microsoft Research, Beijing, China
[3]The University of Newcastle, Callaghan, Australia
[4]Microsoft Azure, Redmond, USA
chenjohn@buaa.edu.cn,hongyu.zhang@newcastle.edu.au
{xian.yang,qlin,v-hadon,Yong.Xu,v-lihao,kay,fgao,zhangxu,Dang.Yingnong,dongmeiz}@microsoft.com

ABSTRACT

With the rapid growth of cloud service systems and their increasing complexity, service failures become unavoidable. Outages, which are critical service failures, could dramatically degrade system availability and impact user experience. To minimize service downtime and ensure high system availability, we develop an intelligent outage management approach, called AirAlert, which can forecast the occurrence of outages before they actually happen and diagnose the root cause after they indeed occur. AirAlert works as a global watcher for the entire cloud system which collects all alerting signals, detects dependency among signals and proactively predicts outages happened anywhere in the whole cloud system. We analyze the relationships between outages and alerting signals by leveraging Bayesian network and predict outages using a robust gradient boosting tree based classification method. The proposed outage management approach is evaluated using the outage dataset collected from a Microsoft cloud system and the results confirm the effectiveness of the proposed approach.

CCS CONCEPTS

• **Software and its engineering → Software maintenance tools**; *Maintaining software*; *System administration*;

KEYWORDS

Outage prediction, outage diagnosis, cloud system, system of systems, service availability

ACM Reference Format:
Yujun Chen[1,2,*], Xian Yang[2], Qingwei Lin[2], Honyu Zhang[3], Feng Gao[4], Zhangwei Xu[4], Yingnong Dang[4], Domgmei Zhang[2], Hang Dong[2], Yong Xu[2], Hao Li[2], Yu Kang[2] . 2019. Outage Prediction and Diagnosis for Cloud Service Systems. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313501

[*] Work done during internship at Microsoft Research.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313501

1 INTRODUCTION

A typical cloud system is a system of systems, providing different services such as networking, storage, computation, security and management. Cloud providers, such as Microsoft, Amazon, Google, and IBM, aim at fast delivery of computing resources in a dynamically scalable and virtualized manner [3, 25]. Many cloud services could be hosted in a cloud system and each service itself is a large and complex system that consists of many components. Failures would happen in a complicated system due to frequent updates of components, changes in operation environment, online repairs, and mobility of devices, [10, 12, 27]. Failures could dramatically degrade system availability and lead to bad user experience. To manage failures, various system monitors and alerting tools are deployed at different places of a cloud service system to detect if a service performs well or not.

In cloud systems, outages are critical system failures that can lead to system unavailability. When an outage is detected, the management tool is expected to automatically notify, mitigate, and diagnose the outage. In literature, there is a large amount of work on predicting and diagnosing failures in a large and complex system such as a data center, grid system, and defense system [14, 15, 20, 30]. Such an ability can help prevent potential disasters and minimize damages caused by system unavailability. However, they only consider a single system and ignore the related systems that could have an impact on the prediction results. In this work, we are interested in predicting and diagnosing outages of a cloud system. A typical cloud system contains many sub-systems (i.e., services), each of which consists of many interconnected components. Each component has its own monitors that regularly check the runtime status of the component. Signals from components reflect different aspects of system health status, such as individual cloud resource, node/data center traffic volume, response latency, temperature, and power consumption. While component-level alerting signals are useful, it is important to have a global watcher for the entire cloud system which understands the topology, resiliency models and dependencies of the system. After connecting all the signals, the global watcher can proactively monitor and correlate service health issues across the whole system without manually defined rules.

In our work, we propose an intelligent outage management tool, called AirAlert, as a global watcher of the whole cloud system. AirAlert collects all alerting signals across the whole cloud system, and utilizes them to diagnose and predict outages. The outages we

try to predict come from two levels: component-level and service-level. These two concepts are hierarchical, where service-level outage consists of various corresponding component-level outages. The service-level outage prediction can help locate the suspicious behavior in the general system, also, locating which component is responsible for the outage can better alleviate the cost of diagnosing and debugging. To diagnose where outages come from, we adopt a Bayesian network approach and investigate dependency relationships between signals and outages. We also construct predictive models to achieve robust outage prediction.

In our work, we are investigating a large-scale Microsoft cloud system. As there are a large number of altering signals, it is hard to know beforehand the dependencies among the signals and their relationships with outages. Therefore, a comprehensive global watcher, specially designed for diagnosing and alerting failures, is needed. In our approach, Bayesian network is used to build the signal dependencies from historical failure statistics. For outage prediction, many algorithms can be used for constructing a prediction model, such as time-series forecasting method (e.g., auto-regressive moving average), rule-based methods (e.g., frequent event set mining [6, 24]), and supervised machine learning methods [8, 17]. However, traditional methods are not capable of handling outage prediction in a cloud service system as they ignore interactions among different system components. There are several challenges in constructing an effective predictive model. The foremost one is that we have very imbalanced datasets due to the limited number of outage cases. Therefore, we apply sampling techniques to preprocess the data and then use the gradient boosting tree based classification method for robust model construction.

Our approach has been evaluated using a set of outage data collected from the Microsoft cloud system over an one year period. We compare the AirAlert approach against other methods and obtain good performance in outage prediction. Also, we show that AirAlert can help outage diagnosis and present several real-world cases.

The contributions of the paper are as follows:

- We propose an outage prediction and diagnosis approach for a cloud service system. Our model takes into consideration multiple related services and components in the cloud system.
- We can infer causal relationships among outages and alerting signals, which can help us understand where the outage is from and which components work cooperatively.
- We can predict whether a component or service will have an outage in the near future, which would potentially help engineering teams alleviate the influence of outages as quickly as possible.
- We have evaluated the proposed approach using an one-year outage dataset collected from a Microsoft cloud system, and the results confirm the effectiveness of the proposed approach.

The remaining sections of this paper are organized as follows. Section 2 describes the related work of the study. Section 3 presents the proposed outage prediction and diagnosis approach and Section 4 describes the experiments. Section 5 concludes our work.

2 RELATED WORK

Failures occurred in online systems could degrade system performance and impact user experience. They are often defined as 'an event that occurs when the delivered service deviates from correct service' [1]. Traditional work on failure management for online systems deals with failures after they have happened. Recently, there are increasing efforts on proactively predicting failures to prevent potential disasters or minimize damages [20]. For internet service provider networks, there are intensive work on failure predictions (e.g., [11, 13, 21, 22]). For example, in [30], switch failures in data center networks are predicted using signals reflecting switch system current status and data containing curated historical hardware failure cases. Input signals of the predictive model are mostly system event logs, such as changes in configuration, interface, device working mode and operational maintenance. Similarly, in [9], network failures in data centers are predicted using data recording network errors related to device and link failures. There are also many studies on investigating the failure characteristics [19, 23, 29] of high performance computing systems (HPCs). In [18], failures occurred in a 350-node cluster system is predicted using various system reliability, availability and service ability (RAS) logs containing health related events. In [7], a failure prediction framework for HPCs is designed by exploring correlations among failures and forecasting the time-between-failure. In [26], a Bayesian network based fault diagnosis tool is developed for managing a defense system-of-systems (SoS), which collects signals from a sensor, network, command and control systems. Bayesian network is used to find the root causes of system failures by using the network topology of the whole SoS. In our work, we use the Bayesian network to find the relationships among alerting signals and outages of cloud systems and then we use XGBoost model for prediction.

The work in [10] studies proactive failure management for cloud systems. It uses Bayesian models and decision trees to proactively predict failure dynamics. Failures are regarded as abnormal signals detected by monitors and failure prediction is tackled from an anomaly detection point of view. Different from their works which focus on predicting failures based on raw signals, our work uses the failure signals from monitors (which we call, *alerting signals*) to predict critical failures (which we call, *outages*). In current real-time system, outage prediction is an important issue that bothers many systems. The reason lies in several aspects: outages are critical system failures that could lead to severe consequences; outages occur without a significant alerting signals pattern, which makes it harder to predict; and the scope of impacting signals is a complex process to define.

3 PROPOSED APPROACH

In this section, we introduce our proposed approach, which is called AirAlert. We first give definitions for our task. Then, we introduce the Bayesian network for generating relationships between alerting signals and outages as well as the gradient boosting tree classification method for outage prediction.

3.1 Definitions

Alerting Signal A single alerting signal A^i can be represented by $A^i = [A^i_1, A^i_2, ..., A^i_T]$, where each component A^i_t indicates the

strength of this signal at time $t \in (1, T)$. For all alerting signals, we denote a multivariate time series of length T as $A = [A_1, A_2, ..., A_T]' \in \mathbb{R}^{D \times T}$ for all signals, where for each $t \in \{1, 2, ..., T\}$, and $A_t = [A_t^1, A_t^2, ..., A_t^D] \in \mathbb{R}^D$ represents the observations of all the alerting signals at time t.

Outage Sequence The outage sequence is a binary time series of length T as $O = (O_1, O_2, ..., O_T) \in \mathbb{R}^T$, where for each $t \in \{1, 2, ..., T\}$, $O_t \in \{1, 0\}$ indicating whether an outage happens at time t or not.

Outage Prediction. Outage prediction at time t is a classification task. We want to maximize the probability $Pr(O_t|\boldsymbol{\mathcal{A}}_t)$, which is predicting accuracy of O_t given the input feature $\boldsymbol{\mathcal{A}}_t$. The input feature $\boldsymbol{\mathcal{A}}_t = [A_t^1, A_t^2, ..., A_t^m]$ are the m alerting signals used for prediction, where $m \leq D$.

3.2 Bayesian network for outage diagnosis

A cloud system is an online system of systems, where the occurrence of an outage is associated with a combination of alerting signals. For example, if a particular web application encounters an outage, it is normally resulted from multiple failures occurred in different services such as networking, hardware and DNS server, rather than from one single failure of the web application component . To be more specific, when several alerting signals are observed, the chance of having an outage depends on all alerting signals. However, it is hard to enumerate all possible combinations of alerting signals and determine which set of signals is related to an outage. To better model the problem, we used a Bayesian network inference method to detect the relationship between alerting signals and outages. Here, we use the FCI-algorithm [5] to infer our Bayesian network. The fundamental idea of FCI algorithm is to build a directed acyclic graph (DAG), where each node X_i represents an alerting signal or an outage based on the causal Markov assumption. The FCI algorithm can be used for the connectivity inference and orientation determination. In this paper, we use it to generate the connectivity between the alerting signals and outages rather than infer the direction.

In our method, the conditional dependence between the alerting signal and outage given a set of other alerting signals are obtained by calculating the Pearson correlation. The influence of the conditional signals needs to be regressed out first. Then, the Fisher's z-transform is performed as in [5]. For example, given the time series sequence of an alerting signal A_i and an outage O_i, and the conditional set only contains the alerting signal A_{i2}, the correlation between A_i and O_i given A_{i2} is:

$$\begin{aligned} r &= \frac{cov(A_i|A_{i2}, O_i|A_{i2})}{\sigma_{A_i|A_{i2}}\sigma_{O_i|A_{i2}}} \\ &= \frac{\sum_{t=1}^{T}(A_{i|i2} - \bar{A}_{i|i2})(O_{i|i2} - \bar{O}_{i|i2})}{\sqrt{\sum_{t=1}^{T}(A_{i|i2} - \bar{A}_{i|i2})^2}\sqrt{\sum_{t=1}^{T}(O_{i|i2} - \bar{O}_{i|i2})^2}}. \end{aligned} \tag{1}$$

Then, Fisher-z transform is used as follows to generate the score for testing the significance of the correlation value:

$$z = \frac{1}{2}\ln(\frac{1+r}{1-r}). \tag{2}$$

z is the correlation score for the alerting signal A_i and outage O_i give A_{i2}. The significant test of z is then used to check whether the independence assumption can be accepted that A_i is independent of O_i given A_{i2}.

The FCI-algorithm is an approximation method to obtain the network structure by calculating conditional dependence. It adopts a recursive inference process and uses the Fisher-z conditional independence test to capture all the independence possibilities among all alerting signals and outages. A bootstrap method for stable result generation is used here [16]. After using FCI-algorithm, the skeleton of the causal network among alerting signals and outages can be obtained. In our work, the Bayesian network mainly works as a diagnositic tool to infer the relationship between the alerting signals and the outage. We can also use it to select the most relevant features and feed them as the inputs of the outage prediction model.

3.3 Gradient boosting tree for outage prediction

In practical online systems, conventionally outage prediction is performed using rule-based methods [20]. Despite the complex pattern of outage, each component or service only focuses on their own alerting signals. For each component or service, engineers will simply set up a monitor to predict outage by examining whether the strength of the alerting signal exceeds a certain threshold θ:

$$P(t) = \begin{cases} 1 & \text{if} A_t > \theta \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where A_t is the observation of alerting signals reported from a certain component at time t, and the value of θ for each team is set based on human knowledge. Such rule-based outage prediction method is normally effective for in-system outages, as this type of outages only caused by malfunction of a component .

In cloud systems, outages are often caused by cross-service malfunctions simultaneously. Thus, using a single signal of a component or service will not be very effective. As a result, we use the alerting signals A_t at time t as the input feature for outage prediction. At a certain time slot t, several alerting signals $\boldsymbol{\mathcal{A}}_t$ will be collected to predict whether the outage will happen at this time. To construct a robust predictor, we use the gradient boosting tree based model (XGBoost) [4]. XGBoost is fundamentally a regression tree that has the same decision rule as the decision tree model. But the model ensembles a set of classification and regression trees (CART). In the CART tree, each node is one of the alerting signals. The prediction result is the sum of scores predicted by K decision trees, as:

$$\hat{y}_t = \sum_{k=1}^{K} f_k(\boldsymbol{\mathcal{A}}_t), f_k \in F \tag{4}$$

where $f_k(\cdot)$ is the score from the kth tree, F is the function space containing all regression trees, and $\hat{y}_t$ contains prediction results at time t. The XGBoost is optimized to achieve the best prediction results by minimizing the following loss:

$$\mathcal{L} = \sum_{t=1}^{T} l(y_t, \hat{y}_t) + \lambda \sum_{k=1}^{K} \Omega(||f_k||) \tag{5}$$

where the first term $\sum_{t=1}^{T} l(y_t, \hat{y}_t)$ is the summation of cross-entropy loss measuring whether the classification model and features can

well perform for the prediction task across all timestamp. The second term $\lambda \sum_{k=1}^{K} \Omega(||f_k||)$ penalizes the tree boosting parameters f_k in the model as described in [28].

4 EXPERIMENTS

In this section, we carry out extensive experiments using data collected from six representative services across tens of data centers over one year period from the Microsoft cloud system. Due to the privacy policy of Microsoft, we deliberately masked sensitive data throughout this paper. We first show the results of predicting outages at component and service levels, where AirAlert has two prediction modes: AirAlert Related and AirAlert Full. In the AirAlert Related mode, only outage dependent signals found in the Bayesian network are used for prediction, while in the AirAlert Full mode all signals are used for prediction. Then, we give case studies to show how our work can help to diagnose outages and find which team to handle the outage.

4.1 Outage prediction

In this subsection we investigate the performance of predicting the component-level outages and also the service-level outages. These experiments aim at investigating the performance of our method for predicting outages of different complex levels. The component-level outage is the outage coming from a specific component . As a service would contain multiple components, the service-level outage is more complicated and heterogeneous.

4.1.1 Experiment Setups. The investigated component-level outages come from three most representative components: the Storage Location, Physical Networking and Storage Streaming component. The service-level outages come from three most representative services: the Web Application, Cloud Network and the Microsoft Cloud System Operation Service. We evaluate the method by collecting outages over one year period from tens of data centers. We acquired data at the time step of one hour and obtained over 8,000 samples in total (24hrs*365days). The period that the data covered only has a very small number of outages. For evaluation, we choose the service and component outages that occurred frequently in the last year (that is why we choose three most representative services/components). For other services/components, which rarely have any outages happened, there is no sufficient data to construct a predictive model. As we only have a limited number of outages , classification methods would suffer from the imbalanced data problem. To better deal with the imbalanced data, the **SMOTE** [2] over-sampling strategy is used for generating the training data from system database so that the positive and negative samples in the training phase can be balanced.

Given alerting signals across the cloud system, we predict whether a component or service outage will happen. We use precision, recall and F1 as the evaluation metrics. Five different outage prediction methods are compared, which are:

- Simple Spike: This is a rule-based outage prediction method as described in Equation 3, where the threshold θ is predefined by domain knowledge and could vary across different components or services . If the strength of the alerting signal is larger than θ, an outage will happen.
- Support Vector Machine (SVM): SVM is commonly used in classification problems. In this experiment, all the alerting signals are served as input features for the SVM classifier with linear kernel.
- Penalized Logistic Regression (PLR): PLR is commonly used in feature selection and prediction with the introduction of sparse constraints. In this experiment, all alerting signals are used as input features for the PLR model.
- AirAlert Related: It first applies the Bayesian network method to find the most relevant alerting signals for outage prediction, which are signals in the Bayesian network directly connected to the outage. Then, we use these selected signals for outage prediction using the Xgboost classification method.
- AirAlert Full: It is based on the XGBoost classifier. Different from 'AirAlert Related', this approach uses all alerting signals as input features.

AirAlert Related and AirAlert Full are two modes of our method. Users can choose which mode they want to use based on their needs. When they want to use all signals for prediction, they can choose the full mode. However, for the dataset with limited number of samples but large number of signals, full mode would easily return an overfitted model. Therefore, we provide the AirAlert Related mode, where only the signals selected by Bayesian network are fed into the gradient boosting tree model.

4.1.2 Results. The results for the component-level and service-level prediction are shown in Table 1 and 2 respectively.

Table 1 shows the outage prediction results at the component-level. Here, we select three most representative and frequently occurred component-level outages for performance evaluation, which are related to Storage Location, Physical Networking and Storage Streaming. All five different prediction methods are compared in terms of precision, recall and F1. We can see that their performance results are quite similar. Simple spike, which is a straightforward rule-based outage prediction method, can achieve 100% recall in some cases for component-level outage prediction. We will later show its performance in predicting service-level outages, which are more complex and involving many heterogeneous signals. Unlike other methods using the whole feature set for prediction, AirAlert Related only uses the Bayesian network selected features. With reduced feature set size, the performance can be still maintained. This observation shows that Bayesian network can help us find the most representative and dependent alerting signals. Other signals which are not found to be highly related to outages in the network would not significantly improve the performance.

Table 2 shows the outage prediction results at the service level. Again, three most representative and frequently happened outages are selected, which are from Website Application, Cloud network and Microsoft Cloud System Operation. Different from the results in Table 1, we can see that the performance of different prediction methods varies greatly. Among them, Simple Spike obtains very low precision and recall. This is because the service-level outage is more complex than the component-level outage and simple rule-based method cannot work for complicated cases. The outage from a single service can be heterogeneous. For example, outages coming

Table 1: Comparison of different methods for component-level outage prediction.

	Outage (Storage Location)			Outage (Physical Networking)			Outage (Storage Streaming)		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Simple Spike	61.65	100.00	76.28	73.71	67.71	70.58	61.52	100.00	76.18
PLR	70.02	92.71	79,78	67.72	83.33	74.72	63.23	91.67	74.84
SVM	65.65	95.83	77.92	63.13	88.54	73.71	58.62	88.64	70.57
AirAlert Related	65.31	100.00	79.01	63.33	98.95	77.25	62.34	100.00	76.80
AirAlert Full	71.11	100.00	83.17	69.07	100.00	81.71	63.75	98.99	77.86

Table 2: Comparison of different methods for service-level outage prediction.

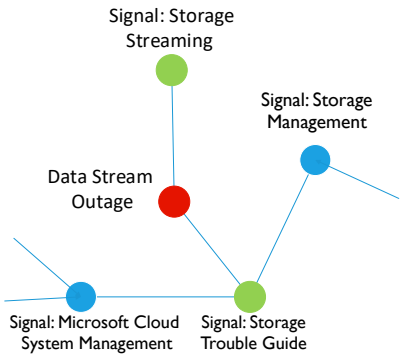
	Outage (Website Application)			Outage (Cloud Network)			Outage (Microsoft Cloud System Operation)		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Simple Spike	5.73	11.83	7.72	4.47	67.74	8.39	7.27	29.03	11.63
PLR	61.18	54.17	57.46	26.27	60.52	36.64	20.36	35.17	25.79
SVM	66.41	88.54	75.89	6.89	88.42	12.78	26.90	22.50	24.50
AirAlert Related	92.18	85.63	88.78	62.08	47.65	53.92	72.40	77.96	75.08
AirAlert Full	82.75	76.74	79.63	75.93	67.07	71.22	72.59	50.15	59.32

from the "website application" service may result from network or hardware failures.

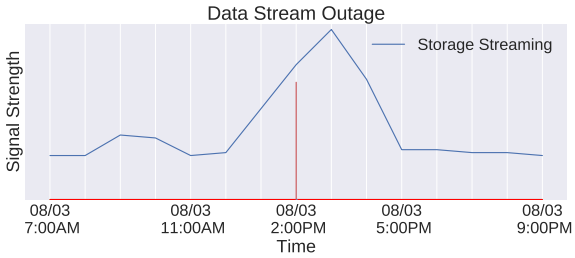
From Table 2 we can further observe that different from Simple Spike, PLR and SVM, AirAlert Related and AirAlert Full have more stable results across different outages. This observation tells us that XGBoost is more robust than others. Consistent with Table 1, the performance of AirAlert Related using Bayesian network selected features has similar performance as AirAlet Full, which uses the whole dataset. More specifically, for the first and the third outage, using related alerting signals about 10% gain in both precision and recall can be achieved. With the help of the Bayesian network, not only the most relevant alerting signals can be identified but also the most predictive features can be detected. The most relevant signals are the ones which would directly lead to outages, while the most predictive features are the ones which contribute to the predictive model the most.

4.2 Outage diagnosis

4.2.1 Cases for outage diagnosis. As mentioned in the previous section, Bayesian network can specify the root cause of outages by calculating the conditional dependence. Here, we would like to give two examples of Bayesian network results. Fig. 1 and Fig. 2 show component-level and service-level outage diagnosis results, respectively. In Fig. 1, subplot (a) contains the Bayesian network for diagnosing relevant alerting signals for the outage called 'Data Stream Outage'. In the graph, the studied outage is in red; its directly linked nodes are in green; and the two-hop neighbours are in blue. This figure shows that the 'Storage Streaming component signal' and the 'Storage Trouble Guide component signal' are most relevant to the occurrence of the 'Data Stream Outage'. When we look up records saved in the Microsoft cloud system, we find that engineer team manually diagnosed the causes of the Data Stream outage

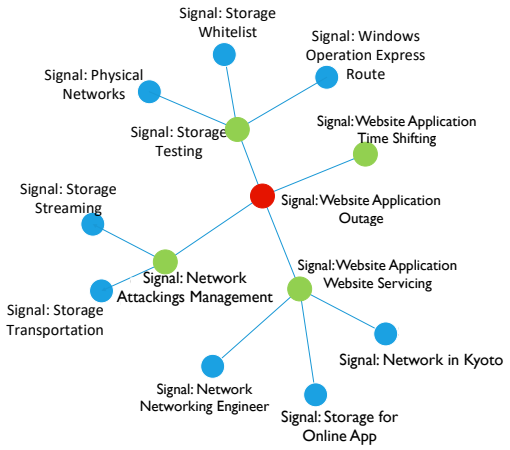


(a) Bayesian network for 'Data Stream Outage'.

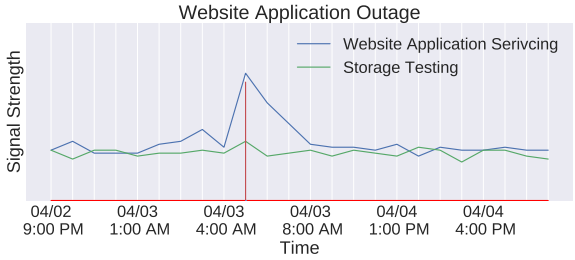


(b) Time series of the 'Storage Streaming' Signal.

Figure 1: Diagnosis results for a component-level outage named 'Data Stream Outage'.



(a) Bayesian network for 'Web Application Outage'.



(b) Time series of the 'Web Application Services' and 'Storage Testing' Signals.

Figure 2: Diagnosis results for a service-level outage named 'Web Application Outage'.

based on their knowledge. They found that the Storage Streaming signal is the top first cause, which is consistent with our findings from Bayesian network. Fig. 1 (b) presents the time-series change for alerting signals, where there was one 'Data Stream Outage' happened at 2:00 PM Aug 3rd. At that time, the related 'Storage Streaming signal' rose significantly, showing the findings in the Bayesian network are quite reasonable.

In Fig. 2, we show an example of diagnosing a service-level outage, called 'Web Application Service Outage'. Fig. 2 (a) tells us how the alerting signals are related to this service-level outage. Comparing Fig. 2 (a) with Fig. 1 (a), we can see that Fig. 2 (a) has more interconnected nodes, which indicate that the service-level outage is more complex than component-level outages and thus has more relevant alerting signals across the system. The signals that directly linked to the 'Website Application Service Outage' are 'Storage testing signal', 'Web Application Time Rotation signal', 'Network attacking Management signal' and 'Web Application Website Servicing signal'. From the records of the Microsoft cloud system, The engineering team has manually identified a large number of Web Application Service outages that are related to 'Web Application Website Servicing', which matches our results well. In Figure 2 (b), we also show the changes of some alerting signals over time, where one 'Web Application Service Outage' happened at 5:00 AM April 3rd. At that time, both 'Web Application Website Servicing signal' and 'Storage testing signal' had a significant jump, which is consistent with the observation in Figure 2 (a). These two examples show that our work on Bayesian network can be a useful outage diagnosis tool that helps the engineering team locate the root cause of outages.

4.2.2 Cases for outage assignment. In a real cloud system, determining which team to handle each outage is a non-trivial task. If we correctly predict the outage to be the 'TeamA Outage', the system will automatically assign the *TeamA* team to handle this outage. Here, we would like to give two examples to show the potential of using our method to assign outages. In these examples, the cloud system first wrongly assigns an outage to a team and then reassigns the outage to the correct team sometime later. However, AirAlert correctly predicted which part of the cloud would have the outage and indicated the correct responsible team who should handle it. There are two instances of the 'Location Servicing Outage' happened on June 7th at 16:00 and July 19th at 16:50, respectively. When we look at the operation record in the Microsoft cloud system, this outage was first assigned to the 'Windows Cloud System Live Site' team and then transferred to the 'Location Servicing' team. If we use our prediction and diagnosis approach, we find that we can correctly predict in advance that an outage would happen in 'Location Servicing Outage'. From the Bayesian network diagnosis results, we can find alerting signals related to 'Location Servicing Outage'. By investigating the time series changes of those related signals, we can see that they increased obviously before these two instances of 'Location Servicing Outage'.

5 CONCLUSION

In this paper, we studied the outage diagnosis and prediction problem in cloud systems. We proposed an intelligent outage management method called AirAlert. In contrast to the previous work focusing on dealing with failures in a single service, we work on outages that would come from many parts of the whole cloud system, which is a system of systems. AirAlert works as a global watcher of the cloud system by collecting alerting signals across the whole system. The dependence relationships among the alerting signals and outages are discovered based on the Bayesian network method. AirAlert can diagnose where the outages come from to help us identify the root causes of outages. For outage prediction, we use the gradient boosting tree based model and compare the results from both component-level and service-level outage predictions. To the best of our knowledge, this is the first work that uses alerting signals across the whole cloud system to diagnose and predict outages coming from different levels.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful and constructive comments. The work was supported by the National Natural Science Foundation of China (Project No. 61702107 and 61828201).

REFERENCES

[1] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions*

on dependable and secure computing 1, 1 (2004), 11–33.

[2] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[3] Junjie Chen, Xiaoting He, Qingwei Lin, Yong Xu, Hongyu Zhang, Dan Hao, Feng Gao, Zhangwei Xu, Yingnong Dang, and Dongmei Zhang. 2019. An Empirical Investigation of Incident Triage for Online Service Systems. In *Proceedings of the 41st ACM/IEEE International Conference on Software Engineering*. to appear.

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[5] Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. 2012. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics* (2012), 294–321.

[6] Carlotta Domeniconi, Chang-Shing Perng, Ricardo Vilalta, and Sheng Ma. 2002. A classification approach for prediction of target events in temporal sequences. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 125–137.

[7] Song Fu and Cheng-Zhong Xu. 2007. Exploring event correlation for failure prediction in coalitions of clusters. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*. ACM, 41.

[8] Song Fu and Cheng-Zhong Xu. 2010. Quantifying event correlations for proactive failure management in networked computing systems. *J. Parallel and Distrib. Comput.* 70, 11 (2010), 1100–1109.

[9] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. 2011. Understanding network failures in data centers: measurement, analysis, and implications. In *ACM SIGCOMM Computer Communication Review*, Vol. 41. ACM, 350–361.

[10] Qiang Guan, Ziming Zhang, and Song Fu. 2012. Ensemble of bayesian predictors and decision trees for proactive failure management in cloud computing systems. *Journal of Communications* 7, 1 (2012), 52–61.

[11] Guenther Hoffman and Miroslaw Malek. 2006. Call availability prediction in a telecommunication system: A data driven empirical approach. In *null*. IEEE, 83–95.

[12] Qingwei Lin, Ken Hsieh, Yingnong Dang, Hongyu Zhang, Kaixin Sui, Yong Xu, Jian-Guang Lou, Chenggang Li, Youjiang Wu, Randolph Yao, et al. 2018. Predicting Node failure in cloud service systems. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM, 480–490.

[13] Qingwei Lin, Jian-Guang Lou, Hongyu Zhang, and Dongmei Zhang. 2016. iDice: problem identification for emerging issues. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. IEEE, 214–224.

[14] Qingwei Lin, Hongyu Zhang, Jian-Guang Lou, Yu Zhang, and Xuewei Chen. 2016. Log clustering based problem identification for online service systems. In *Proceedings of the 38th International Conference on Software Engineering Companion*. ACM, 102–111.

[15] Debra Greenhalgh Lubas. 2017. Department of defense system of systems reliability challenges. In *Reliability and Maintainability Symposium (RAMS), 2017 Annual*. IEEE, 1–6.

[16] Sara Magliacane, Tom Claassen, and Joris M Mooij. 2016. Ancestral causal inference. In *Advances in Neural Information Processing Systems*. 4466–4474.

[17] James W Mickens and Brian D Noble. 2006. Exploiting availability prediction in distributed systems. *Ann Arbor* 1001 (2006), 48103.

[18] Ramendra K Sahoo, Adam J Oliner, Irina Rish, Manish Gupta, José E Moreira, Sheng Ma, Ricardo Vilalta, and Anand Sivasubramaniam. 2003. Critical event prediction for proactive management in large-scale computer clusters. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–435.

[19] Ramendra K Sahoo, Mark S Squillante, Anand Sivasubramaniam, and Yanyong Zhang. 2004. Failure data analysis of a large-scale heterogeneous server environment. In *Dependable Systems and Networks, 2004 International Conference on*. IEEE, 772–781.

[20] Felix Salfner, Maren Lenk, and Miroslaw Malek. 2010. A survey of online failure prediction methods. *ACM Computing Surveys (CSUR)* 42, 3 (2010), 10.

[21] Felix Salfner and Miroslaw Malek. 2007. Using hidden semi-Markov models for effective online failure prediction. In *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*. IEEE, 161–174.

[22] Felix Salfner and Steffen Tschirpke. 2008. Error Log Processing for Accurate Failure Prediction.. In *WASL*.

[23] Bianca Schroeder and Garth Gibson. 2010. A large-scale study of failures in high-performance computing systems. *IEEE Transactions on Dependable and Secure Computing* 7, 4 (2010), 337–350.

[24] Ricardo Vilalta and Sheng Ma. 2002. Predicting rare events in temporal domains. In *null*. IEEE, 474.

[25] Asher J Vitek and MN Morris. 2012. Service oriented cloud computing architectures. In *UMM CSci Senior Seminar Conference Morris, MN*.

[26] Hongbing Wang, Lei Wang, Qi Yu, Zibin Zheng, Athman Bouguettaya, and Michael R Lyu. 2017. Online reliability prediction via motifs-based dynamic bayesian networks for service-oriented systems. *IEEE Transactions on Software Engineering* 43, 6 (2017), 556–579.

[27] Yong Xu, Kaixin Sui, Randolph Yao, Hongyu Zhang, Qingwei Lin, Yingnong Dang, Peng Li, Keceng Jiang, Wenchi Zhang, Jian-Guang Lou, et al. 2018. Improving service availability of cloud systems by predicting disk error. In *2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)*. 481–494.

[28] Zhixiang Xu, Gao Huang, Kilian Q Weinberger, and Alice X Zheng. 2014. Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 522–531.

[29] Praveen Yalagandula, Suman Nath, Haifeng Yu, Phillip B Gibbons, and Srinivasan Seshan. 2004. Beyond Availability: Towards a Deeper Understanding of Machine Failure Characteristics in Large Distributed Systems.. In *WORLDS*.

[30] Shenglin Zhang, Ying Liu, Weibin Meng, Zhiling Luo, Jiahao Bu, Sen Yang, Peixian Liang, Dan Pei, Jun Xu, Yuzhi Zhang, et al. 2018. PreFix: Switch Failure Prediction in Datacenter Networks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 2.