A Review on Faults and Anomaly Behaviour for Fault Diagnosis and Prediction in the Distributed Network

[1]M. Srinivasa Rao, [2]D. Nagendra Rao, [3]V. Usha Shree, [4]S. Prabhakara Rao
[1]Ph.D Scholar, [2]Professor, [3]Professor, [4]Professor
[1]Department of Electronics and Communication Engineering,
[1]JNTUH, Hyderabad, India.

Abstract: As computer networks become more complex, fault diagnosis has become a difficult task for network operators. In general, an fault in a communication system is always caused by a large amount of alert information. Due to the large amount of information, it is time-consuming and fault-prone to manually identify the root cause. Therefore, automatic malfunction diagnosis of computer networks is an open research problem. Detecting these defects is one of the most challenging tasks in modern computer networks. However, most of the existing fault or anomaly detection methods are generally designed to model normal traffic and then find the variance of that model. However, these techniques do not have labels for identification and suffer from other problems, including many false positives. In this paper, it provides a review of defect and anomaly behaviour for fault diagnosis and prediction in distributed networks. It explores the challenges of identifying various defects and abnormal behaviour in the network. It also provides insight into methodologies for fault diagnosis and forecasting approaches. This review will be an additional benefit for researchers to understand fault and anomaly prediction in distributed networks.

Keywords: **Distributed Network, Fault Prediction, Anomaly Behaviour, Fault Diagnosis.**

1. INTRODUCTION

Over the past decade, the world has experienced rapid development in network applications of various types, and networks have become more advanced in terms of heterogeneity, complexity and volume. Some barriers, such as availability, flexibility and scalability, have affected existing central network management systems, with networks becoming more distributed [1]. There is an important problem with distributed systems that are affected by component failure and are distributed diagnostics. In distributed diagnostics, each business node must keep the correct information about its status as "operational" or "failed" for each component in the system. Many past researches has dealt almost exclusively with fixed faults and fixed state of a single node that takes an algorithm to fully diagnose the system [2], [3]. Even some work has tried to take into account dynamic events, but since no formal model has been developed, the accuracy and evaluation of algorithms has returned to static models.

In recent years, a lot of research has been done to troubleshoot network problems, especially fault diagnosis and management [4]. However, according to recent audit studies, the direction of defect management and diagnosis has been increasingly discussed recently, but has not contributed much to the diagnosis of defects in computer network systems. There are more flaws in space travel, manufacturing, navigation, nuclear industry and hospitals with the growth of larger and more complex systems [5], [6]. Complex systems always have lots of assemblies to work together. Since the emergence of computer networks, there are more and more application systems on the network to share knowledge to improve production efficiency, in such case the network reliability has become an essential issue.

Current computer networks, for example telecommunications networks, have become much larger and more complex. One fault in a network component could cause network operators to report a very large volume of alarms. The explosion of the alarm may result from a "repeated fault" or "multi-service call from a faulty component" or "multiple alarms generated by a single fault device" or "detect and issue a network fault notification by multiple devices at one time", and spread the fault to other network devices causing their failure [7]. Thus, it is a challenge for network operators to identify the root cause quickly and correctly by analyzing this large amount of alarms.

The predicting or detecting intruder and malware infections on the local network is one of the most difficult and modern security issues [1], [8]. Many of the proposed detection methods use fixed rules or reputation methods for detection performance. More modern behavioural technology has been introduced. This anchor technique is very useful, but it was not enough to detect most attacks and malware. In particular, it believes that the most important limit of current technology is as follows. (a) Predictions are made for each connection, not for each user. (b) Classifiers shall be trained and checked only for normal and infected datasets, and (c) The types of attacks and infections reduce the usefulness of the classifier.
The restricted fault correction specification is an enhancement to the characteristics required to solve the diagnostic problem. In the past, many solutions have been offered to find solutions, but no predefined diagnostic algorithm has been formally established to achieve powerful features such as specific accuracy characteristics in the presence of truly dynamic breakdowns and fixes. Current diagnostic and predictive methods such as "Snort" [9] and "Bro" [10] rely on signature-based IDS and there has been an extensive research into behavioural detection methods over the past decade. Among these new methods, the most commonly used are "Anomaly prediction (AP)". Most of these AP systems [11] assume that more than half of the data is normal and detect anomalies by looking for their natural and light deviations.

A key benefit of AP technology [12] is the ability to identify non-malicious problems within the network, such as detecting previously unknown attacks and violating corporate policies. Although widely used, APs are struggling with many problems that weaken its usefulness. First, it is difficult to see the results, so common communication and blending attacks occur. Second, existing exceptions are not necessarily harmful and cause large amounts of false positives. Third, as the nature of the network changes over time, the original model may no longer be usable and faults may occur. Fourth, the AP method generally works with the package, so road stability is poor. The amount of faults caused by these problems tend to be so large that researchers tend to apply different algorithms to the AP output to improve detection. As a result, the AP model tends to maintain ongoing supervision until an acceptable outcome is achieved.

To improve the current situation, it is necessary to focus accurately on the various changes in system behaviour and diagnostics to predict correct faults in dynamic and distributed networks [13]. This must be achieved through much shorter diagnostic crossing times with much higher behavioural change rate than previous events by extending previous studies focused on the classification of defects in the actual application of computer network systems to understand differences.

In the following paper, it presents the significance of faults in network connections in section-2 and the identification of anomaly behaviour in section-3. The methodology for diagnosis of faults and anomaly is presented in section-4, and it the related fault prediction methods in section-5. In section-6, it presents the related works, and in final seection-7 it presents the conclusion.

2. FAULTS IN THE NETWORK

Systems widely distributed are the foundation of current Internet services and continue to grow in popularity and importance. However, the reliability of distributed systems is a major challenge [14]. In addition, many successful systems become heterogeneous due to multiple execution and distributed systems, multiple providers of multiple federal administrative domains are deployed as a result of widespread deployment of networks.

Faults are network events that are causes for malfunctioning [15]. Thus, faults can cause other events. A class of faults which are not themselves caused by other events are named root causes. Faults may propagate across the entire network. It is because that many network objects are dependent on each other, and a fault in one object always causes faults in its depending objects. Fault propagation is one cause of alarm burst. This event, as a special condition in the work of network devices or software, is a key concept in diagnosing faults [16]. An event-related technique or event is assigned as a managed object. Events can be classified as primitive or composite events [3], [17]. Primarily pre-defined events in the system usually occur directly in managed objects. Composite events are conceptual events constructed from primitive or low-level formative events.

In the early days, the diagnosis of network dysfunction was based on professional knowledge and implementation [18]. In accordance with the ISO/OSI model, it can use the network test tool to monitor and measure network parameters in the three bottom layers as "physical", "data link", "network layer", are use in the protocol analyzer on all layers, to collect and manage network information. It uses some test commands like "ping" and " traceroute " for the data link layer to check the broken link.

Event linking is a method that theoretically interprets various events so that events that have the same root cause are grouped together [19]. After the link, the number of event notifications decreases, but the semantic content grows. Thus, the event link, as the most common localization technology, helps network operators dramatically detect the root cause of large information. The most important association types are listed as follows:

- *Reduction*: Reduce alerts, which are multiple events notifying one event at a time.
- *Counting*: Replace a new alarm with a number of specific alerts related to a recurring event.
- *Causal association*: Correlation when there is a cause-effect relationship to the events behind the alarms.
- *Temporal association*: Correlate alarms according to the order or time the alarm is generated. The alarms caused by that fault are likely to be observed in a short time after a sequence or fault occurs.

3. IDENTIFICATION OF ANOMALY BEHAVIOUR

Definition of anomalies depends on the target system, available data and implementation conditions [8]. The nature of distributed systems, where cumulative behaviour is the result of complex actions of multiple nodes, is impossible to test and debug their code separately, as well as to fully understand the overall effect of local actions. Considering the inter-regional route, restoring the BGP session in response to a synthetic but useless route might be the perfect way to address the disadvantages for a router. However, when it is associated with a large number of routers that propagate a potential fault, the overall effect is a large proportion of the routers that are constantly renewing sessions. High levels of regeneration analysis, reminiscent of emerging behaviour create problems in identification and it credibility [11].

User behaviour is defined by all actions and decisions taken by the user during a certain period. Their tasks are changed into packets and flows, which are then captured in the characteristics described earlier in the profile. It allows each profile to describe a user's behaviour with twelve different perspectives, each capturing a different perspective [12]. As time goes by and the user generates more traffic, more and more profiles are produced. The user's behaviour is then defined by all these profiles and their characteristics. However, each feature describes the same data separately and therefore there is no sense to compare each feature with each other. Instead, it proposes to compare each feature in the profile, the same profile has the same feature for this user. The idea is that when the analysis of the same feature is done in conjunction with the profile then the anomalies will arise.

3.1 Challenges in Anomaly Detection

It argues that anomaly detection is more challenging to create a redesigned, unbalanced distribution system-wide system [20] because (i) the source code of each node may not be readily available for testing and (ii) the potential for competing concerns to motivate individual providers. That it maintains his current position and personal part of the configuration.

A. ***Heterogeneity*:** Distributed systems that are widely deployed are often successful because of the open standards and well-defined interfaces that permit multiple implementations. Further, even the software deployed by the same vendor has multiple versions and patch revisions due to the difficulty of instantaneously upgrading all nodes. Moreover, heterogeneity arises from the lack of global coordination in systems operated under multiple administrative domains. The resulting heterogeneity creates a problem for fault prediction because it is difficult or impossible to have the source (or even binary) code for all nodes that is required to achieve spatial and temporal awareness in existing approaches [21]. Thus, the approach should be able to operate using only the existing interfaces.

B. ***Hidden internal state*:** Despite existing business relationships among providers motivates both sides to predict faults, the federated nature of large-scale distributed systems translates to the desire of nodes to keep a large portion of their state and configuration that captures business practices private [22]. This hinders the ability of external entities to identify problems with a node's configuration or software. A solution in this space should only use the well-defined interface and ideally leak no confidential information. It is only in this case that the distributed system operators would have the incentive to participate in the protocol and increase overall system reliability.

C. ***Incremental deployment*:** While attempting to redesign the protocols and the programming interfaces, a solution that has any chance of success has to be incrementally deployable [23]. This means that the approach should not require changes to the existing protocol messages. Also, it should not pose unnecessary requirements on the programming interfaces or require intrusive changes to the existing systems.

D. ***Small Executing Time*:** The approach should be able to explore distributed system state during the short quiescent periods [24]. In addition, it should be able to predict potential faults in a timeframe that is short enough for the operators to take preventive measures or the system to automatically adapt. However, a definite trade-off exists between exploration speed and bandwidth consumed. Exploring systems that have a large amount of traffic will require managing the bandwidth used for exploration so that the system performance is not degraded to undesired levels.

E. ***Long executing Times/Longer inputs*:** The deployed systems it are interested in will potentially be running for months without restarting. This means that the inputs to the system will be long, which further makes achieving path exploration and good coverage difficult [25].

F. ***False positives/negatives:*** As with any approach that performs fault detection and prediction, false positives and negatives represent a potential issue. False negatives can occur if the properties that are checked for are not capable of discerning the faulty state. Regarding the false positives, live execution over the shadow snapshot is evidence of the behaviour that is the result of processing a particular input. However, it is challenging to ensure that the properties themselves are defined in a way that avoids false positives. Also, it is not possible to detect faults that are not checked for. However, automatically inferring system invariants is a substantial challenge [26].

The algorithms that can be used for anomaly detection are varied and include any algorithm that can differentiate between distributions of data. This is the case of One-Class SVM, that has been used for anomaly detection by M. Zhang et al. [12]. It used One-Class Support Vector Machines to detect anomalies. They evaluated their approach on the dataset KDDCUP99 which was created in 1999. The algorithm showed very promising results compared to other methods. There are multiple algorithms for anomaly detection in the past literature. V. Chandola et al. [20] reviewed different types of anomalies, the different fields where anomaly detection is used, challenges of anomaly detection and algorithms that could be used for anomaly detection.

4. FAULTS DIAGNOSIS METHODOLOGY

Fault diagnosis is a new emerging technology on equipment operation, maintenance, began in 60 to 70 years of aviation, aerospace and nuclear engineering [2], [5], [14], [16], it refers to a certain working environment lead to the identification of the nature and causes of the mechanical system a dysfunctional, judge the deterioration state occurring parts, and predict the development trend of the state of deterioration. The purpose of fault diagnosis is: the first is to improve the reliability of equipment operation, to avoid the equipment for the sudden failure and catastrophic damage caused enormous economic losses [22]. The second is the accurate prediction of the running state of the equipment, ensure the maximize the function of the equipment and production efficiency. Third is to provide the technical foundation for the equipment repair system reform, save repair cost and human, material resources. In addition, the technology of fault diagnosis but also for the machine design, manufacture and operation of testing service.

Diagnosis algorithms can use either unicast or multicast communication. The bulk of the work in system diagnosis has assumed a static fault situation [27], [28], i.e., the statuses of nodes do not change during execution of the diagnosis procedure. Some diagnosis algorithms, e.g., [17], [29] allow dynamic failures and repairs to occur, but are only guaranteed to be correct when system status has become stable. One of the diagnosis algorithms in [17] assumes that nodes can fail dynamically, but cannot be repaired during execution of the diagnosis procedure. This approach is suitable in some systems, but is not a satisfactory solution in general. The diagnosis model of [29] considers dynamic failures but requires a centralized diagnosis entity.

Previous work on distributed diagnosis has focused almost exclusively on minimizing the number of tests performed. One interesting result of our work is to show that the goal of minimizing tests and the goal of effectively handling dynamic failures and repairs are directly in conflict. Prior algorithms that minimize the number of tests construct sparse testing graphs and propagate information in reverse direction of tests. The ideal testing property for which these algorithms strive is to have each node tested by exactly one other node at each testing round.

Another related area is that of failure detection. Failure detectors are used to solve higher-level problems such as consensus and atomic broadcast in asynchronous and partially-synchronous systems. To our knowledge, the latest work that considers failures and recoveries is [30]. It presents the existence of nodes that eventually are permanently working or permanently failed is assumed. Since it do not assume existence of such nodes in our model, all nodes are unstable in the terminology. So, the working nodes do not distinguish between unstable and failed nodes. Hence, no evaluation is done of the minimum time an unstable node needs to be in a particular state so that its status is accurately tracked.

4.1 Perceptions of Fault Diagnosis

Network fault diagnosis [31] is the goal of the available time and maximize the network, improve the utilization rate of network equipment, network performance, service quality and safety, simplifying the mixed management under the network environment and reduce network operation cost control, to extend the service life of the network. The essence of fault diagnosis is pattern recognition, because of the diversity and complexity of network equipment and fault form, between the network fault symptom and fault condition is not a simple one correspondence from the fault symptoms set to fault state set is a complex nonlinear mapping. Fault diagnosis is to detect the state information generated operating process of network equipment, extract the sign reflecting the network running status of equipment features from the detected signals, and to identify the state of equipment according to the symptoms and other diagnostic information, find the fault part, find the causes to find fault, puts forward the corresponding measures of troubleshooting to complete the fault diagnosis [14].

With the rapid development of computer network technology, the scale and function of networks is constantly increasing. The increasing importance and complexity of networks led to the development of network fault management as a distinct field, providing support for network administrators with quality services and ensuring that networks work appropriately. Fault diagnosis is a central aspect of network fault management. Since faults are unavoidable in communication systems, their quick detection and isolation is essential for the robustness, reliability and accessibility of the system. In large and complex communication networks, automating fault diagnosis is critical. Because of many factors, including the volume of network information, it is hard to solve network fault problems with traditional tools, rendering intelligent diagnosis a critical method in the process of network fault diagnosis [17].

4.2 Fault Management Activities

Fault management activities are performed through interactions between the fault management service user and the fault management functions. This section describes the fault management activities in terms of operations done within fault management functions and interactions between fault management functions and fault management service users. Depending on the case in which fault management is active, the following five different activities are identified [31].

- *Alarm Surveillance*: includes collection and logging of alarm notification from the network resources.
- *Fault Localisation*: analyses the collected alarm information, detects the root cause of alarm, and notifies the result to the clients of the alarm surveillance.
- *Fault Correction*: restores and recovers the computational objects that represent the resources in which a root cause alarm is detected.
- *Testing Function*: invokes a test capability of a resource object upon a request from the clients of the service. It may also support a test of series of resource objects.
- *Trouble Administration*: enables the reporting of troubles due to fault conditions and the tracking of their status.

Fault diagnosis is a process of finding out the original cause for the received symptoms (alarms). It usually involves three steps [32]:

- Fault detection, an on-line process which indicates that some network objects are malfunctioning according to the alarms reported by those objects.
- Fault localization (also referred to as fault isolation, alarm/event correlation and root cause analysis), a process that proposes possible hypotheses of faults by analyzing the observed alarms.
- Testing, a process that isolates the actual fault from a number of possible hypotheses of faults.

4.3 Diagnosis Methodology

Early fault diagnosis techniques are too simplex to find complex faults and rely too much on the professional experience. Compared with the rapid network developing in scale and amount, early fault diagnosis techniques are poor on collecting information, analyzing data, getting real root causes, and becoming inefficient. Usually, the fault diagnosis in networks is plotted into three sections: information gathering, information analyzing, diagnosing and resolving [33].

A. Information gathering

Information gathering can be divided into three kinds: active, passive, active-passive method. Most active methods of gathering information depend on Simple Network Management Protocol (SNMP) [34]. In SNMP polling model, agents running on the aim network element and a central controller running on a computer are necessary. Central controller sends request for aim status to the agents periodically. Some network management systems use this method, like the "Open View system of HP Company" and the "Net View system of IBM Company". For sending request to agents and agents echoing timely, networks cost mu bandwidth and time on transferring and computing.

Passive gathering SNMP Trap makes the controller monitor the SNMP Trap, without sending anything. So this method is real-time. However, Trap is carried by the User Datagram Protocol (UDP), which can not ensure the quality of transmission. So passive gathering SNMP Trap is easy to lose something important.

B. Information analyzing

Information analyzing is a process in which useful symptom is extracted from fault information, and fault is located, and isolated. It can be divided into two groups: exact inference and approximate inference. Exact inference has following methods: graph reduction, combinatorial optimization, poly tree propagation; approximate inference has following methods: method based on simulation, method based on searching, and transformation method. Transformation method is more important than others two [35].

C. Diagnosing and resolving

Diagnosis is a process that makes certain the location and type of fault. There are three classes: "analytical model-based method", "signal processing-based method" and "knowledge-based method" [36]. Among those methods which are used in the Diagnosis and resolving, the knowledge-based method becomes the primary research filed because of its self-rule and intelligence. The knowledge-based method is divided into many methods and technologies: fault diagnosis based on fault tree, fault diagnosis based on expert system, fault diagnosis based on Fuzzy Logic, fault diagnosis based on artificial neural network, fault diagnosis based on Grey theory, and fault diagnosis based on Bayesian networks [29].

5. FAULTS PREDICTION APPROACHES

In this context, by fault prediction it mean to detect possible sequences of actions which reach states that present inconsistencies that can lead to failures. As these might actually never happen, our prediction is loose with respect to time in that it is not associated to a time window during which the failures could occur. For a thorough discussion on failure prediction [37]. According to literature [1], faults are the adjudged or hypothesized cause of an fault. A failure refers to misbehaviour that can be observed by the user. Given that it is exploring possible actions, these include faults. Therefore, because it wants to avoid failures, i.e., prevent faults that have a visible erroneous state, it refers to various approach as fault prediction.

5.1 Rule-based Approach

Rule-based approach is significantly used in many commercial fault diagnosis products. In rule-based systems, the diagnostic knowledge of a human expert is modelled as rules, which are saved in a knowledge-base. Formally, rules are expressed in form of production rules, e.g. if A then B, where A is called antecedent and B is called consequent. Antecedent is usually the assertion on the frequency and the source of an alarm as well as the values of its properties.

This approach is widely used because human experts' knowledge can be intuitively defined as rules. Furthermore, it does not require profound understanding of the underlying system, which eases developers from domain learning. However, rule-based approach has the following downsides:

- The procedure of knowledge acquisition, which is based upon interviews with human experts, is always time-consuming, expensive and fault-prone. However, some approaches can automatically derive correlation rules based on the statistical data.
- It is unable to learn from experience, therefore the rule-based systems are subject to repeating the same faults.
- It is difficult to maintain because rules frequently contain hard-coded network configuration information.
- It is unable to deal with unseen problems.
- It is difficult to update system knowledge.

5.2 Model-based Approach

In contrast with the traditional rule-based approaches, model-based approaches rely on some sorts of deep knowledge beside the surface knowledge. This deep knowledge is known as system model, which may describe system structures in terms of network elements and the topology, and its behaviours in the process of alarm propagation and correlation [17].

- The system model usually uses an object-oriented paradigm [38] to represent network elements as well as the relationship between them.
- A class is a template for a set of real network elements. All network elements that are instances of one class share the properties defined in that class.
- Each subclass inherits properties from its super class. Therefore, inheritance allows system components to be treated generically regardless of their specific details when they are not relevant.

Due to the use of deep knowledge, model-based approaches are able to address some issues in rule-based systems. The diagnostic knowledge is now easy to maintain since its condition part associates system model instead of hard-coded network configuration [5], [13]. The condition part asserts current network configuration by utilizing predicates referring to the system model. Predicates test the current relationships among system components. Additionally, knowledge in model-based systems can be organized in an expandable, upgradeable and modular fashion by taking the advantage of object-oriented paradigm. Moreover, model-based systems have the potential to solve novel problems [38]. Although model-based approaches are superior to rule-based approaches, they have problems about obtaining models and keeping the models up-to-date.

5.3 Case-based Approach

Contrary to rule-based and model-based systems, case-based systems can learn from past cases to propose solutions for new problems [31]. Here, the knowledge is in terms of cases not rules or models. Besides their ability to learn case-based systems are not subject to changes in network configuration. However, it is a complicated and domain-dependent process to adapt an old case to a new situation. It proposes a technique named parameterized adaption to address this issue. Additionally, the case-based approach may be not used in real-time alarm correlation due to the time inefficiency.

5.4 Decision Tree Approach

A decision tree models an expert's decisions and their possible consequences and can be used to guide a process of diagnosis to reach the root cause. Expert knowledge can be simply and expressively resented by using decision trees [7]. Moreover they have crucial advantage of yielding human-interpretable results, which is important for network operators. However, their applicability is limited due to the dependence on specific applications and the poor accuracy in the presence of noise. A decision tree is usually constructed from data by using the machine learning technique.

In general, rule-based approaches can be used for a simple system which is rarely changed. Model-based systems present an additional system model in relation to rules, which make the superior to the pure rule-based systems but does not make them more attractive due to the difficulty of obtaining and update the model. Although case-based systems are less sensitive to changes in network, they are not suitable for handling real-time alarm correlation. In addition to their own problems, neural networks and decision trees both rely on a long training period and may not work outside the area of training. A two-phase approach for measuring the performance of the cluster based internet services was presented in [39]. The first phase of the methodology employs the fault-injection approach for measuring the impacts of faults on the network performance while the second phase makes use of analytical models to assess the network performance by combining the measurements of first phase and the fault loads. Such a two-phased approach lets the evaluator study how the servers respond to various design-related decisions, rate of faults and other factors.

6. RELATED WORKS

Distributed network fault diagnosis model based on Bayes classifier proposed a distributed network fault diagnosis system model framework based Agent, the model for Bayesian classification theory of promotion [29]. Model prior knowledge and observation data together, which greatly improved the diagnostic performance of the system. Model uses a certain status check and verification strategies to ensure their own safety and the safety of Agent Communication provides a common framework for network fault diagnosis system. The key aspect of network fault management is the process of a fault classification, by which it concludes the details of a failure from a set of tested failure indications.

Faults are unavoidable and cause network downtime and degradation of large and complex communication networks. The need for fault management capabilities for improving network reliability is critical to rectify these faults. Current communication networks are moving towards the distributed computing environment enabling these networks to transport heterogeneous multimedia information across end to end connections. An advanced fault management system is thus required for such communication networks. Fault Management provides information on the status of the network by locating, detecting, identifying, isolating, and correcting network problems thereby increasing network reliability.

K. Xu et al. [40] also used NetFlows to analyze the traffic. Their system created a cluster for each IP in the current time window. Clustering was based on the "srcIP". For each cluster, the system computed the normalized entropy of "scrPort", "dstPort" and "dstIP" and used it as a feature vector to represent clusters. Then the system applied behaviour classification scheme to classify each sample in its behavioural class. It was similar to our proposal in the sense that they also saw the information about Source Ports and Destination Ports as important for anomaly detection.

M. V. Mahoney et al. [41] also inspected TCP flags but based on individual packets. The proposed NETAD algorithm [13] built nine models to identify anomalies in nine subsets of packets. Packets were split into subsets based on TCP flag in the packet and on the port. The algorithm achieved 66 detections out 185 with only 20 false alarms. The area of smart networks has been the subject of good amount of research and review recently because of the concept of computational intelligence, which is incorporated into smart networks [14]. Because of computational intelligence, smart networks are capable of detecting their faults and classifying them [35]. The two fundamental techniques that can be used to predict future system behaviour by operating on the source code are model checking and symbolic execution.

Model checking involves exploring system states by executing enabled actions (e.g., timers, message handlers, local actions) in each encountered state. For distributed systems, the aggregate state is composed of the states of participating nodes. Model checkers also require a harness that can introduce faults, e.g., broken connections, node failures, reordered messages, etc. The

CrystalBall [37] instantiates the state machines from live, consistent node checkpoints. It can determine safety and liveness violations spanning multiple nodes and it was used to find bugs in systems implemented in the MACE [42] distributed systems framework.

The CrystalBall [37] goes one step further in that it can proactively predict inconsistencies that can occur in a (Mace-based) running distributed system due to unknown programming faults, and effectively prevent them. MODIST [43] goes one step further than MACE in that it is capable of model checking unmodified distributed systems. One could potentially use MODIST to orchestrate state space exploration across a cluster of machines in an isolated (non-deployed) scenario. A general issue with model checking techniques is the exponentially large set of potential system states.

7. CONCLUSION

Fault diagnosis in networks has made great progress in common fault detecting and management. The paper presents a review on faults anomaly behaviour for Fault Diagnosis in modern computer networks. It is primarily difficult to make distributed systems reliable and a key step in doing so is the ability to automatically predict faults. Each method of fault diagnosis in networks relies on one or more theories, which determinates the application of method. It discusses the challenges in anomaly behaviour prediction in related to the system faults and states changes. It put an insight on the methodology of fault diagnosis and its management activities. It also discuss the various approaches can be applied to perform the fault prediction and to improving the robustness of fault diagnosis algorithm for the future research and practical applications.

REFERENCES

[1]. W. He, C.-Qiang Yu, Guo-Hui Zhou, Zhi-Jie Zhou, Guan-Yu Hu, "Fault Prediction Method for Wireless Sensor Network Based on Evidential Reasoning and Belief-Rule-Base", IEEE Access, Vol. 7, 2019.

[2]. Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, M. Mukherjee, "A Survey on Fault Diagnosis in Wireless Sensor Networks", IEEE Access, Vol. 6, pp. 11349 - 11364, 2018.

[3]. E. Procopio Duarte, Andrea Weber, Keiko V. O. Fonseca, "Distributed Diagnosis of Dynamic Events in Partitionable Arbitrary Topology Networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 23(8), 2012.

[4]. X. Jia, Y. Jiang, J. Zhu, "Link fault protection and traffic engineering in hybrid SDN networks", IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 853 - 858, 2018.

[5]. Y. Tian, Xiangyu Liu, "A deep adaptive learning method for rolling bearing fault diagnosis using immunity", Tsinghua Science and Technology, Vol. 24(6), 2019.

[6]. A. Zakari, Sai Peck Lee, Ibrahim Abaker Targio Hashem, "A Community-Based Fault Isolation Approach for Effective Simultaneous Localization of Faults", IEEE Access, Vol. 7, 2019.

[7]. M. Chen, A.X. Zheng, M.I. Jordan, and E. Brewer, "Failure Diagnosis Using Decision Trees", International Conference on Autonomic Computing (ICAC), New York, NY, May 2004.

[8]. W. Dong, Luyao Luo, Chun Chen, Jiajun Bu, Xue Liu, Yunhao Liu, "Post-Deployment Anomaly Detection and Diagnosis in Networked Embedded Systems by Program Profiling and Symptom Mining", IEEE Transactions on Parallel and Distributed Systems, Vol. 27(12), 2016.

[9]. M. Roesch et al., "Snort: Lightweight intrusion detection for networks", In Lisa, Vol. 99(1), pp. 229–238, 1999.

[10]. V. Paxson, R. Sommer, S. Hall, C. Kreibich, J. Barlow, G. Clark, G. Maier, J. Siwek, A. Slagell, D. Thayer et al., "The bro network security monitor", 2012.

[11]. Z. Lan, Z. Zheng, Y. Li, "Toward Automated Anomaly Identification in Large-Scale Systems",IEEE Transaction on parallel and distributed system, Vol. 21(2), pp.174-187 February 2010.

[12]. M. Zhang, B. Xu, and J. Gong, "An anomaly detection model based on one-class svm to detect network intrusions", IEEE 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN), pp. 102–107, 2015.

[13]. C. Wallace, Patrick Benavidez, Mo Jamshidi, "Predicting Fault Behaviors of Networked Control Systems Using Deep Learning for Mobile Robots", 14th Annual Conference System of Systems Engineering (SoSE), 2019.

[14]. D. Cabrera, Fernando Sancho, Jianyu Long, René-Vinicio Sánchez, Shaohui Zhang, Mariela Cerrada, Chuan Li, "Generative Adversarial Networks Selection Approach for Extremely Imbalanced Fault Diagnosis of Reciprocating Machinery", IEEE Access, Vol. 7, 2019.

[15]. Huang, Bingkui Chen, Liming Xiao, Yan Ran, Genbao Zhang, "Cascading Fault Analysis and Control Strategy for Computer Numerical Control Machine Tools Based on Meta Action", IEEE Access, Vol. 7, 2019.

[16]. A. Zafar, B.Wajid, and B. A. Akram, "A hybrid fault diagnosis architecture for wireless sensor networks", in Proc. IEEE Int. Conf. Open Source Syst. Technol. (ICOSST), pp. 7-15, Dec. 2015.

[17]. Q. Zheng, Y. Qian, M. Yao, "A network event correlation algorithm based on fault filtration", Springer PRICAI 2006 - Trends in Artificial Intelligence, vol. 4099, pp. 864– 869, 2006.

[18]. P. Godefroid, Nils Klarlund, and Koushik Sen, "DART: Directed Automated Random Testing", In Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '05), 2005.

[19]. S. N. Ananthan, Surya Santoso, "Universal model-based fault location for improved system integrity", IET Generation, Transmission & Distribution, Vol. 13(8), 2019.

[20]. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey", ACM computing surveys (CSUR), Vol. 41(3), pp. 15, 2009.

[21]. J. Neuzil, O. Kreibich, and R. Smid, "A distributed fault detection system based on IWSN for machine condition monitoring", IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 1118-1123, May 2014.

[22]. W. G. Fenton, T. M. McGinnity, L. P. Maguire, "Fault diagnosis of electronic systems using intelligent techniques: a review", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 31(3), pp. 269–281, 2001.

[23]. L. Lin, Li Xu, Shuming Zhou, Sun-Yuan Hsieh, "The Extra, Restricted Connectivity and Conditional Diagnosability of Split-Star Networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 27(2), 2016.

[24]. R. Majumdar and Koushik Sen, "Hybrid Concolic Testing", In Proceedings of the 29th International Conference on Software Engineering (ICSE '07), pp. 416–426, 2007.

[25]. C. Cadar, Vijay Ganesh, Peter M. Pawlowski, David L. Dill, and Dawson R. Engler. "EXE: Automatically Generating Inputs of Death",13th ACM Conference on Computer and Communications Security (CCS '06), Alexandria, Virginia, USA, 2006.

[26]. M. Yabandeh, Abhishek Anand, Marco Canini, and Dejan Kostic, "Almost-Invariants: From Bugs in Distributed Systems to Invariants", Technical Report NSL-REPORT-2009-007, EPFL, 2009.

[27]. Y. Gao, X. Zhou, "The design of network fault diagnosis system based on PNN", 2nd International Conference at the Future Computer and Communication (ICFCC), 2010.

[28]. D. Wu, Q. Yang, F. Tian, D. X. Zhang, "Fault Diagnosis Based on K-Means and PNN", IEEE 3rd International Conference on Intelligent Networks and Intelligent Systems (ICINIS), pp. 173–176, 2010.

[29]. Yuan, X. Zhao, and L. Yu, "A distributed Bayesian algorithm for data fault detection in wireless sensor networks", in Proc. IEEE Int. Conf. Inf. Netw. (ICOIN), pp. 63-68, 2015.

[30]. Y. Nakayama, D. Hisano, T. Kubo, Y. Fukada, J. Terada, A. Otaka, "TDD-Based Rapid Fault Detection and Recovery for Fronthaul Bridged Network", IEEE Communications Letters, Vol. 22(3) pp. 498 - 501, 2018.

[31]. Y. Yu, X. Li, X.Leng, L. Song, K. Bu, Y. Chen, J. Yang, L. Zhang, K. Cheng, X. Xiao "Fault Management in Software-Defined Networking: A Survey", IEEE Communications Surveys & Tutorials, pp. 1 - 1, 2018.

[32]. T. Sipola, A. Juvonen, and J. Lehtonen, "Dimensionality reduction framework for detecting anomalies from network logs", Engineering Intelligent Systems, vol. 20, 2012.

[33]. Burnim and Koushik Sen, "Heuristics for Scalable Dynamic Test Generation", Technical Report UCB/EECS-2008-123, EECS Department, University of California, Berkeley, Sep 2008.

[34]. Simple Network Management Protocol, http://en.wikipedia.org/wiki/ Simple Network Management Protocol.

[35]. M. Barakat, F. Druaux, D. Lefebvre, M. Khalil, O. Mustapha, "Self-adaptive growing neural network classifier for faults detection and diagnosis", Neuro computing, Vol. 74(18), pp. 3865–3876, 2011.

[36]. A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "A meta-analysis of the anomaly detection problem", arXiv preprint arXiv:1503.01158, 2015.

[37]. M. Yabandeh, Nikola Kneževi´c, Dejan Kosti´c, and Viktor Kuncak, "Crystalball: Predicting and preventing inconsistencies in deployed distributed systems", In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI '09), 2009.

[38]. T. Huang, H. Sethu, and N. Kandasamy, "A new approach to dimensionality reduction for anomaly detection in data traffic", IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 651–665, 2016.

[39]. Qader, M. Adda, "Network Faults Classification Using FCM", 17th International Conference on "Distributed Computer and Communication Networks (DCCN-2013), 2013.

[40]. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Reducing unwanted traffic in a backbone network", SRUTI, vol. 5, pp. 9–15, 2005.

[41]. V. Mahoney, "Network traffic anomaly detection based on packet bytes", In Proceedings of the 2003 ACM symposium on Applied computing, pp. 346–350, 2003.

[42]. C. Killian, James W. Anderson, Ryan Braud, Ranjit Jhala, and Amin Vahdat. "MACE: Language Support for Building Distributed Systems", In Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '07), San Diego, CA, June 2007.

[43]. J. Yang, Tisheng Chen, Ming Wu, Zhilei Xu, Xuezheng Liu, Haoxiang Lin, Mao Yang, Fan Long, Lintao Zhang, and Lidong Zhou, "MODIST: Transparent Model Checking of Unmodified Distributed Systems", In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI '09), Boston, MA, April 2009.