Research Proposal

Single Pane Alerting and Monitoring for Enhanced Response: Telstra Case Study

By Muhammad Aditya Hilmy (1262771)

The University of Melbourne

School of Computing and Information Systems

Faculty of Engineering and Information Technology

Supervisors: Dr Christine Rizkallah, Prof Christopher Leckie

Table of Contents

1 Problem Statement

Telstra operates a large number of heterogeneous network devices and servers. With so many interconnected devices, failure is bound to happen. The company implements redundancy to keep the systems running in case of failures. However, some failures will have almost no impact on the performance of the system. As a result of this redundancy, failures can go undetected. If unresolved, such failures can manifest over time, degrading performance slowly until it reaches a critical point, and outages can happen.

Currently, human operators monitor various performance metrics and will escalate an issue if they notice potential failures. With plenty of devices to monitor and high volume of metrics to process, escalating alerts manually has proven to be difficult in forecasting potential failures. Telstra has attempted to automate this alerting by thresholding: an alert will be raised if the metric value reaches a certain lower or upper limit. Deciding this threshold, however, is not straightforward. Setting a static threshold will not work well because the values fluctuate over time, unless the threshold is set reasonably high or low. For instance, traffic volume will differ during working hours and midnight, or holidays and weekdays. Furthermore, the defined threshold may not reflect the devices' optimal operating parameters. As a consequence, false negatives or positives can occur.

To forecast potential problems, we can use past data to capture the patterns of normal behaviour, and raise an alert when the current behaviour deviates from the past. This is called anomaly detection [1].

However, even with anomaly detection, discovering the root cause of a problem remains a challenge. Once an anomaly is detected, human operators will have to decide whether the anomaly is benign, and review various metrics and alerts to pinpoint the root cause. With many things that can go wrong, multiple alerts can be issued, and they may be related to the same root cause. As an example, Telstra had instances where thousands of alerts were

raised within seconds, and none of them were helpful to reveal the root cause: fibre optic cables were inadvertently cut. To help human operators to quickly identify the root cause, we can employ *alert correlation* to infer the relationship between different alerts and present them in a more helpful way [2].

This project aims to explore anomaly detection techniques to detect abnormal behaviours in the metrics, and review alert correlation methods to group together relevant anomaly alerts to help human operators discover the root cause more effectively.

2 Literature Review

To address the problem, we review two main areas that could potentially be the solution, namely time series anomaly detection and alert correlation.

First, we explore time series anomaly detection as a method of detecting abnormal behaviour to forecast potential issues. We start this section by outlining the types of anomaly detection methods and proceed to mention some related works in the domain. While there are numerous anomaly detection frameworks in the literature, we present the ones that are normally used in network monitoring scenarios. We also mention some state-of-the-art frameworks to explore novel approaches to the problem.

Second, we explore alert correlation methods as a method of presenting more relevant alerts to the human operators to reduce their workload. We present the types of alert correlation methods and mention some relevant contributions in each type.

2.1 Time-series anomaly detection in network monitoring

Anomaly detection relies on performance metrics received from a variety of devices to capture past patterns in the metrics and detect when pattern deviates from the past behaviour [1]. Its ability to detect even the slightest deviations allows it to discover

potential problems before it becomes too severe. This would help the operations team to resolve issues before they cause an incident.

Methods of time series anomaly detection are divided into three types: proximity-based, prediction-based, and reconstruction-based [3]. Proximity-based methods rely on a distance measure to compute similarity between data points, where data points far from the others are considered anomalies [3]. However, this method requires prior knowledge on the duration of anomalies [3], which is difficult to infer in a highly dynamic network traffic characteristic. In prediction-based methods, past time series data is used to fit a model, and the model is used to predict future data points [3]. When, at a particular time, the actual data is significantly different than the predicted data, it is considered anomalous. In reconstruction-based methods, a model is trained to capture the characteristics of patterns and used to reconstruct the original data [3]. Anomaly is detected when the reconstructed data is far different than the original data.

Several works have been done to explore the use of anomaly detection for network monitoring. In 2002, Barford et al. explored the use of signal analysis to detect anomalies from traffic flow data [4]. Similarly, in 2006, Lakhina et al. propose a framework of detecting anomalies using Principal Component Analysis (PCA) [5]. In 2009, Barford et al. propose a novel approach using *active probing* [6]. Here, an algorithm will detect anomalies in the network paths, and will decide which path to probe to localise the anomalies [6]. These approaches, however, rely on predefined statistical methods to detect anomalies, and may not be adaptable to deal with more complex traffic characteristics.

Geiger et al. proposed the use of generative adversarial networks (GAN) called TadGAN to detect time series anomalies, which uses reconstruction error to calculate anomaly [3]. The authors highlighted that this method offers a robust anomaly score calculation and can reduce false positives. However, GAN is computationally expensive. Therefore, it is challenging to deploy GAN to detect anomalies in real time. To do that, we need to resort to methods for streaming data. Ahmad et al. proposed a method known as Hierarchical

Temporal Memory (HTM) for detecting anomaly in such data [7]. It can detect spatial and temporal anomalies, as well as resilient to temporal drift. Although, TadGAN outperforms HTM during benchmarking [3].

We can combine the two previous methods to create a robust detection system. HTM can be used to analyse metric streams in real-time to forecast imminent potential problems (e.g. sudden drop in traffic). Meanwhile, TadGAN can run in the background to detect anomalies that manifest more slowly (e.g. minor device misconfiguration).

2.2 Alert correlation

Alert correlation allows network operators to better understand alerts coming from heterogeneous sources [2]. It groups together related alerts to help operators discover the root cause while reducing the number of benign alerts [2], thereby allowing them to focus on actionable alerts. There are numerous alert correlation methods that have been proposed in the literature [2]. Salah et al. categorises alert correlation methods into three: similarity-based, sequential-based, and case-based [2].

Similarity-based methods evaluate the intrinsic attributes of individual alerts, using similarity measures to determine whether several alerts are related [2]. While this technique has proven to be effective, it does not consider the causal relationships between alerts [2]. In a heavily interconnected and heterogeneous network where one failure could lead to another, causality relationships may need to be considered to create a higher-level understanding of alerts.

Sequential-based methods attempt to use causality relationships to group alerts, working under the assumption that one incident can lead to the next [2]. One major advantage of these methods is their scalability [2]. Because these methods look at the sequence of events as opposed to individual attributes, they can correlate alerts across heterogeneous sources. For example, similarity-based methods may have trouble to deduce that a power failure is related to latency increase, since those alerts have different attributes.

Sequential-based methods, however, can infer that power failure causes some edge routers to fail, thereby increasing latency. Although, these methods have one major drawback: they may create plenty of false alerts [2]. In which case, false positive detection techniques need to be evaluated to alleviate the issue.

Case-based methods rely on existing knowledge base that captures well-defined scenarios, which can be populated by a human expert, or inferred using machine learning techniques [2]. However, with many things that can go wrong, it is difficult to create an exhaustive list of scenarios.

There are other relevant works in the literature. Costa et al. propose an intelligent alert system for telecommunication companies which can correlate alerts and deduce the root cause automatically [8]. Holub et al. present an approach to correlate alerts at run-time for large volumes of log data [9]. Ramaki et al. devise an alert correlation technique involving Bayesian networks that can forecast multi-step attacks.

3 Methodology

This section outlines the methodology that will be used in conducting this project.

First, we will study the specific monitoring requirements that Telstra has and establish evaluation criteria. Second, we will evaluate anomaly detection and alert correlation techniques that can satisfy the evaluation criteria. Third, we will implement a small-scale proof-of-concept and deploy it to monitor a subset of metrics collected in the Internal DNS subsystem of Telstra network. Finally, we will evaluate the proof-of-concept against the established evaluation criteria.

We plan to evaluate the proposed system as follows.

1) Time series data from several metrics will be collected.

2) Anomalies will be manually labeled before the evaluation based on past incidents and expert judgements of the Telstra team. This results in the *ground truth dataset*.
3) The unlabeled ground truth dataset will be classified by the anomaly detection system.
4) Accuracy, precision, recall, and F1-score will be computed to evaluate the anomaly detection system performance.

4 Timetable and Plan

This section outlines the approximate timeline and plan of this research project.

Start date	End date	Task
28 Feb 2024	15 Mar 2024	Writing the research proposal
14 Mar 2024	14 Mar 2024	Visit to Telstra Global Operations Centre
18 Mar 2024	29 Mar 2024	Requirements elicitation and synthesis
01 Apr 2024	24 May 2024	Creating high-level designs and deriving evaluation techniques, proof-of-concept implementation, and evaluation
27 May 2024	03 Jun 2024	Writing the final report

5 References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection,” *ACM Comput Surv*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[2] S. Salah, G. Maciá-Fernández, and J. E. Díaz-Verdejo, “A model-based survey of alert correlation techniques,” *Computer Networks*, vol. 57, no. 5, pp. 1289–1317, Apr. 2013, doi: 10.1016/j.comnet.2012.10.022.

[3] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, “TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks,” in *2020 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2020, pp. 33–43. doi: 10.1109/BigData50022.2020.9378139.

[4] P. Barford, J. Kline, D. Plonka, and A. Ron, “A signal analysis of network traffic anomalies,” in *Proceedings of the second ACM SIGCOMM Workshop on Internet measurment - IMW ’02*, New York, New York, USA: ACM Press, 2002, p. 71. doi: 10.1145/637201.637210.

[5] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 219–230, Aug. 2004, doi: 10.1145/1030194.1015492.

[6] P. Barford, N. Duffield, A. Ron, and J. Sommers, “Network Performance Anomaly Detection and Localization,” in *IEEE INFOCOM 2009*, IEEE, Apr. 2009, pp. 1377–1385. doi: 10.1109/INFCOM.2009.5062053.

[7] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, “Unsupervised real-time anomaly detection for streaming data,” *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017, doi: 10.1016/j.neucom.2017.04.070.

[8] R. Costa, N. Cachulo, and P. Cortez, “An Intelligent Alarm Management System for Large-Scale Telecommunication Companies,” 2009, pp. 386–399. doi: 10.1007/978-3-642-04686-5_32.

[9] V. Holub, T. Parsons, P. O’Sullivan, and J. Murphy, “Run-time correlation engine for system monitoring and testing,” in *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, New York, NY, USA: ACM, Jun. 2009, pp. 9–18. doi: 10.1145/1555312.1555317.