Proceedings of CECNet 2021
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA210459

Automated Contextual Anomaly Detection for Network Interface Bandwidth Utilisation: A Case Study in Network Capacity Management

Esmaeil ZADEH[a*], Stephen AMSTUTZ[b], James COLLINS[b], Craig INGHAM[b], Marian GHEORGHE[a] and Savas KONUR[a]
[a]*Department of Computer Science, University of Bradford, UK*
[b]*Xalient Holdings Limited, UK*

Abstract. We present a contextual anomaly detection methodology utilised for the capacity management process of a managed service provider that administers networks for large enterprises. We employ an ensemble of forecasts to identify anomalous network traffic. Stream of observations, upon their arrival, are compared against these baseline forecasts and alerts generated only if the anomalies are sustained. The results confirm that our approach significantly reduces false alerts, triggering rather more accurate and meaningful alerts to a level that could be proactively consumed by a small team. We believe our methodology makes a useful contribution to the applications enabling proactive capacity management.

Keywords. contextual anomaly detection, ensemble of timeseries forecasts, data gathering, machine learning, alerting

1. Introduction

The ability to effectively maintain the performance of a network relies on careful management of the capacity of its constituent paths. Anomalous behaviours can potentially result in losses to the business – in both revenue and long-term reputation [1]; i.e., an unusually overutilised path will increase latency and/or create packet-loss resulting in degradation of the user experience.
Traditionally capacity management would be accomplished by monitoring interface traffic and setting up static threshold alerting (i.e., point anomaly [2]) based on a percentage of the configured bandwidth. While this global outlier detection approach can be used to capture large global anomalies, it is ill-suited for detecting seasonal anomalies [1,3] and can lead to a high frequency of alerts, especially during busy periods making it difficult to identify genuine issues within the noise.

An alternate and more effective approach is contextual anomaly detection [3]. Chandola et at. [2] define the term as where a data point is only anomalous in a specific context; i.e. given time context, an interface traffic might be anomalous in weekends but not in weekdays. It is also suggested that modelling the structure of the data and using it

* Corresponding Author. Email: e.habibzadeh1@bradford.ac.uk

for anomaly detection is one of the techniques that is usually applied to time series data with seasonality. Rather than a single forecast model, an ensemble of forecasts where each model can be optimised to address a very specific problem at hand is widely accepted for its improved performance [4,5].

Herein we present a contextual anomaly detection methodology for the capacity management process of a managed service provider that administers networks for large enterprises. Taking into account the trend and seasonality, this approach sets dynamic thresholds based on the variations from the forecasts, and therefore tends to significantly reduce false alerts generated to allow a much smaller team to respond to them. One of the challenges in this case study was accurately generating a small number of meaningful alerts (i.e., eliminating false positives and false negatives), and ensuring these alerts were delivered in a timely manner, ideally within 15 minutes. We discuss the investigations into how data granularity and smoothing functions play a role and can help. We also consider algorithms and parameters used for alert dampening.

The rest of the paper is organised as follows: Section 2 briefly overviews the existing related work; Section 3 describes our methodology; Section 4 presents experimental results; Section 5 summarises the findings; and Section 6 concludes with future work.

2. Related Work

The existing research in anomaly detection is valuable and quite abundant [1,2,6,7,8,9]; nevertheless, the field cannot claim maturity yet, due to the lack of an overall, integrative framework to understand the nature and different manifestations of its focal concept, the anomaly [6]. Hochenbaum et al. [1] argue that detection of anomalies in time series with multimodal distribution, seasonality and/or underlying trend is non-trivial; the authors highlight that a large body of anomaly detection techniques, e.g., the Three-Sigma (i.e., $3.\sigma$) rule, can potentially capture global anomalies, but is not applicable for data with seasonality characteristics mentioned above. They propose detection techniques that involve decomposition of time series into trend and seasonality, followed by point anomaly detection applied to the noise component. Chandola et at. [2] conduct an extensive survey on anomaly detection; they classify anomalies into three types: a) global (point), b) contextual (conditional), and c) collective; they also broadly classify techniques on contextual anomaly detection into two main categories:

- reducing the problem to a global anomaly detection problem, and
- modelling the structure in the data and using it to detect contextual anomalies, which is typically applied to time series data. They provide and discuss, among others, auto-regressive models (i.e. ARMA and ARIMA) that have been widely applied to various application domains including statistics, financial markets, security, networking and graphs.

Ensemble forecasting, however, is widely accepted for its improved performance. Bates and Granger [4] suggest the idea of using multiple forecasts rather than choosing a perfect or even significantly superior forecast model as a way to improve forecast performance. They argue that any forecast model nearly always contains some useful independent information; as such, it is not a wise procedure to discard any of them if the objective is to make as good a forecast as possible.

3. Methodology

Figure 1 illustrates our methodology and the data journey, from the point where it is acquired, to the point where it is evaluated against forecasts and generates alerts.

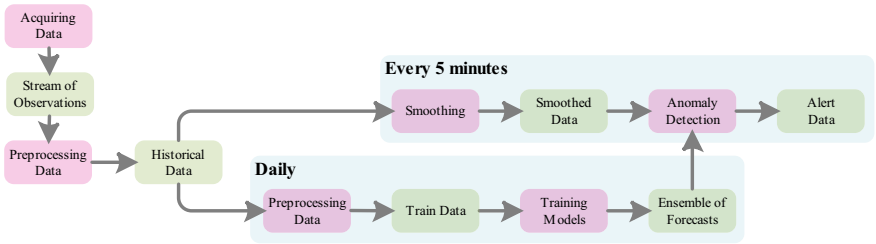

Figure 1. Methodology and data journey block diagram.

Data is sourced from a selection of network devices via a collocated proxy agent. These devices securely report on a regular basis flow records (metadata about each IP conversation traversing a device such as a router, switch, or host) to a proprietary cloud-based agent, storing the records in a cloud data storage. The data here, however, is quite huge for various reasons, but mainly due to its fine granularity (i.e., per second) and that it contains a lot more dimensions and metrics than one would normally require for a certain analytics task at hand. Here, we give focus on interface bandwidth utilisation data.

Per second data is an invaluable asset and sometimes crucial, e.g., for monitoring and analysing min/max network traffic periods for capacity planning; however, it is less suitable for pattern recognition algorithms due to the presence of excessive amount of irregular roughness. As part of data pre-processing, we downsample and take moving average to smooth the data as required. It is worth mentioning, however, that unlike moving average, downsampling tends to reduce granularity, leading to less visibility.

Here, we are particularly interested in detecting contextual anomalies; but we need a mechanism (e.g., a model) to help identify what should be considered normal within certain confidence levels, given the time context. We use past data to learn past behaviours, e.g., seasonality patterns and/or long-term trends, and based on which fit an ensemble of forecast models [10] to predict the future.

It is important to note that granularity of forecast data should be greater than or equal to that of the training data; i.e. given a 5-minute granularity of training data, this suggests 5, 10, 15, … minutes for the forecast granularity. Otherwise, the fitted models will be less confident on predicting values for sub-intervals.

The last stage of the methodology involves the anomaly detection algorithm, which evaluates the stream of actual observations against their respective forecasts and flags them, if significantly deviated, as:

- low anomaly (i.e., underutillised) if it is below the lower boundary
- high anomaly (i.e., overutillised) if it is above the upper boundary

We take high/low anomalies as disparate classes and treat them independently. The annotated observations are then fed to alerting engine, which in turn generates alerts on breach. Combination of contextual anomaly detection and ensemble forecasting alone tends to significantly reduce false alerts. We also define algorithms to help eliminate duplicate alerts and suppress an alert being triggered multiple times before being cleared.

4. Experimental Results

In this section, we demonstrate how our methodology works and its effectiveness in generating timely and more meaningful alerts. In particular, we briefly look into the rationale for using EWM (Exponentially Weighted Mean) as opposed to SMA (Simple Moving Average) to smooth our data. We then demonstrate how ensemble forecasting can help address two common network issues more reliably. Throughout we use real life ingress/egress interface traffic.

4.1. SMA vs EWM

We use moving average to smooth out irregular variations in interface traffic. All these functions apply to a subset of data, sliced by a moving window, with a side-effect of introducing a lag time, proportional to the window size. That is, smoother outputs can only be achieved at the cost of longer lags and therefore increased resolution times. We look into the two common moving average functions (i.e. SMA and EWM) to find out which can better help achieve reasonably smoothed data while minimising the lag times.

Figure 2 confirms that both functions tend to produce very similar smoothed outputs as long as variations are slow and less significant; even the output of SMA_30m does not look too different, despite using a half-size window.

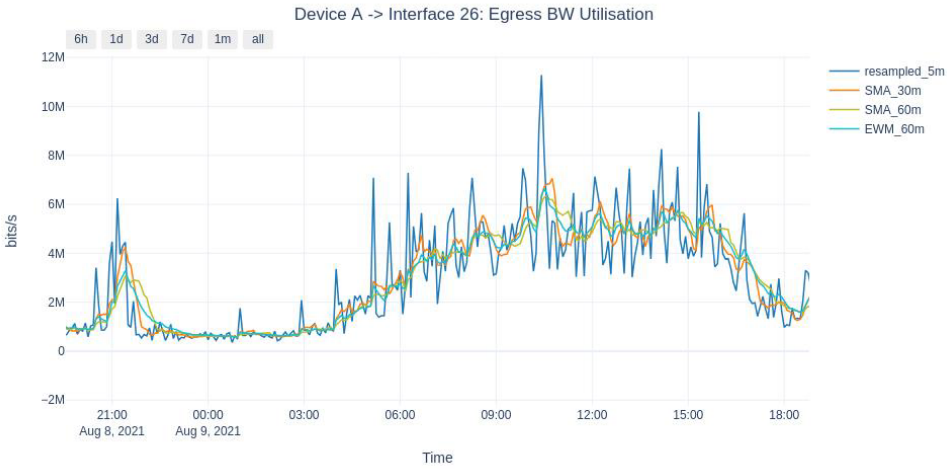

Figure 2. SMA vs EWM: in response to slow.

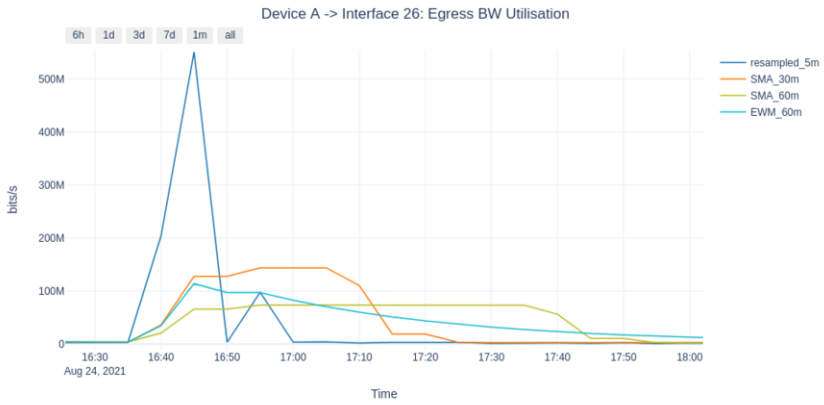

Figure 3. SMA vs EWM: in response to rapid variations.

However, in the presence of rapid changes that are also significant, SMA and EWM tend to behave quite differently. As illustrated in Figure 3, in response to a sharp spike, SMA produces much longer lasting pulses, albeit lower in amplitudes. We also observe that SMA_60m is slower (i.e., longer lag) and left behind the most. EWM, on the other hand, rises and falls so quickly; in fact, despite having double window size, it tends to perform better than SMA_30m, producing smoothed outputs with less distortion. Nevertheless, one may argue that EWM tends to introduce far longer tails; this, however, is very unlikely to have serious implications since the tail dies exponentially.

It is crucial to realise that the distortions due to these smoothing functions can, sometimes, manifest themselves as false persistency, leading to false positive alerts, whereas the original signals with narrow spikes would probably have no chance to trigger any alerts no matter how high in amplitude.

4.2. Ensemble Forecasting

Figure 4 shows actual traffic versus the forecast data, achieved using a single forecast model. Apart from daily and weekly seasonalities, it is also apparent that a daily scheduled job is performed at a specific time (~3-4am).

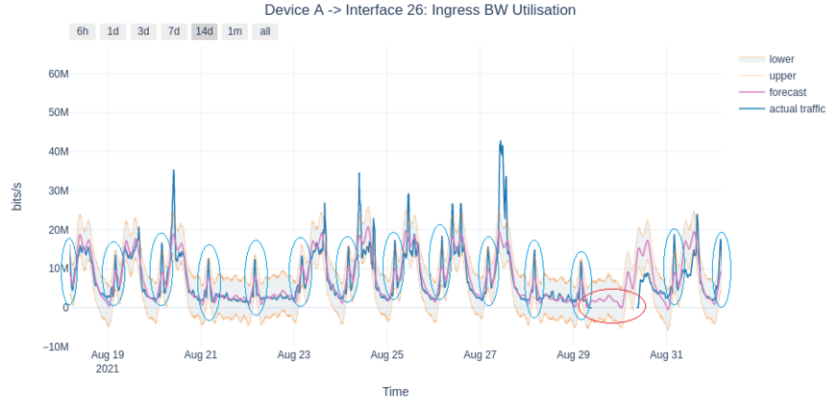

Figure 4. Actual traffic vs forecasts.

At a glance, the model appears to be a reasonably good fit, following major seasonality patterns quite well. However, we realise that the model is partially impaired and ineffective, with two major drawbacks as follows:

- It performs poorly and is less confident around daily scheduled jobs, suggesting way too wide uncertainty intervals to allow for detection of missed jobs! (Figure 4, marked by blue ellipses).
- Also interface failures, e.g. disconnections, or when the interface traffic is zero, will go unidentified since uncertainty intervals include zero values; i.e. no traffic at all are not considered unusual! (Figure 4, marked by red ellipse).

We employ an ensembleforecasts to address these issues and improve overall performance as follows.

4.2.1. Alert on missed scheduled jobs

As mentioned above, the uncertainty intervals, in particular around daily scheduled jobs, are way too wide. It is therefore highly likely that a missed job will go unidentified since there is not enough margin between model's lower bound and the level of dropped traffic due to the absence of a job.

With this problem in mind, we optimize the model such that it is more confident and follows the daily scheduled jobs more accurately. Figure 5 illustrates the new model, which confirms improved margins between presence and absence of a scheduled job; i.e. in the event of a missed job, the dropped traffic will be well below the lower bound and for long enough to allow for timely detection.

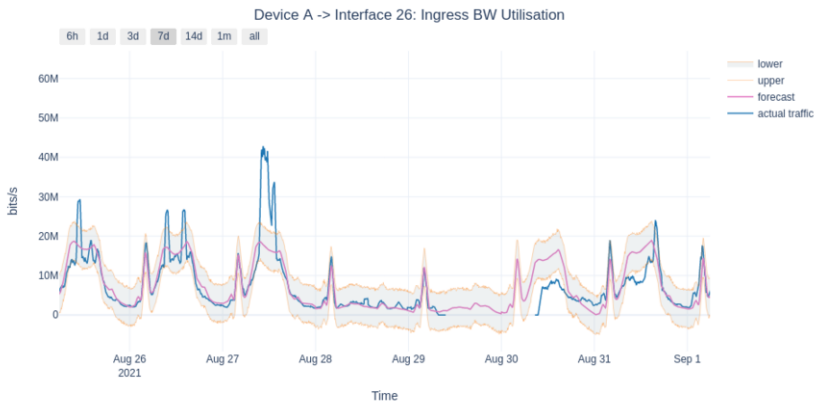

Figure 5. Improved forecast model for daily scheduled jobs

4.2.2. Alert on interface failure

It is not uncommon for an interface traffic to drop to very low levels during evenings and weekends, hence zero is within the expected range. However, due to the type and nature of interfaces used in this study, no traffic at all for quite long periods[1] would be abnormal and timely detection of such cases will be desirable. To achieve this, we apply a log transform prior to fitting the model, which helps to dramatically increase the margin and distinguish zero traffic from the rest quite easily (see Figure 6).

[1] Recall that we aggregate and smooth the traffic data over 60-minute moving windows.

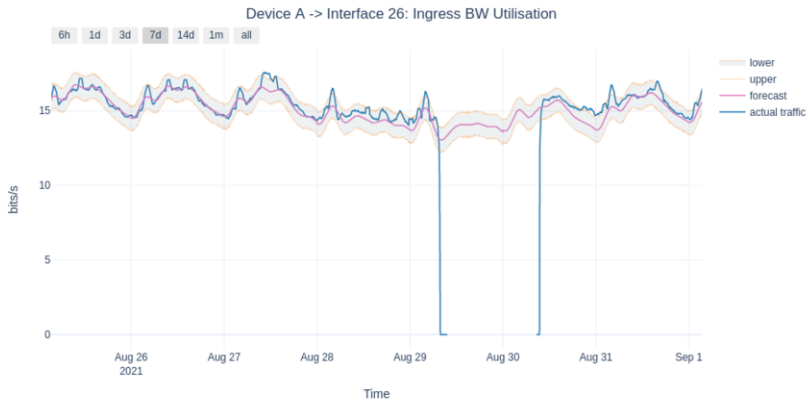

Figure 6. Improved forecast model for interface failure and zero traffic

4.3. Alert Dampening

To reduce false positives, a distinction is drawn between anomalies and alerts. An alert will only be generated when there is a sustained anomaly which is determined using the following parameters:

- Persistent anomalies: where we see *n* consecutive anomalies.
- Impersistent anomalies: where we see *n* anomalies across a specific window size (number of samples).

In addition, once an alert has been generated, we suppress further alerts until there has been a period of stability. This period is determined using the following parameter:

- Alert cleared: where we see *n* consecutive non-anomalous samples.

5. Discussions

Careful selection of the moving average function enabled the smoothed data to better represent the original data resulting in greater accuracy in both the detection and persistency of anomalies, without reducing detection time:

- improved detection: EWM_60m was more likely to reflect an anomaly than SMA_60m due to its higher amplitude.
- quicker detection: EWM_60m raised as quickly as SMA_30m and much quicker than SMA_60m resulting in quicker detection (i.e. shorter lag).
- reduction of false persistency: EWM_60m fell quicker than both SMA_60m and SMA_30m resulting in reduced persistency. This would in turn create less consecutive anomalies therefore reducing the likelihood of an alert being raised.

Using an ensemble of forecasts (versus a single forecast model) improved the accuracy in the detection of anomalies by:

- reducing the uncertainty interval where important anomalies could otherwise be missed (under-utilisation), or

- increasing the uncertainty interval where false-positives are more likely due to bursty traffic (over-utilisation).
- in the case of under-utilisation we found an improvement in detecting the absence of small but regular traffic (scheduled jobs) in some cases from a 10% likelihood of detection to a 100% likelihood of detection.

And finally, utilising alert dampening parameters reduced the quantity of anomalies being escalated to alerts, or duplicate alerts being raised.

6. Conclusions

The exhibition of heavy seasonality in interface bandwidth utilisation is apparent; as such, global anomaly detection techniques are not effective and can lead to too many false alerts. We have, instead, taken a contextual anomaly detection approach together with ensemble forecasts.

Stream interface traffic, upon their arrival, are evaluated against forecast values and flagged anomalous if they are significantly deviating from predicted values. Anomalies get marked as alerts if they satisfy sustainability rules. We also discuss algorithms used to prevent duplicate alerts and suppress any alerts being raised multiple times before they are declared cleared.

The results show that the methods presented can meaningly improve the alert generation process both in quantity and quality to a level that could be proactively consumed by a small team. We believe our methodology makes a useful contribution to the applications enabling proactive capacity management.

The next move is to consider multi-variate (i.e., latency, loss, utilisation) timeseries where channels are correlated and can inform/share useful knowledge about each other.

Acknowledgements

The authors acknowledge the Innovate UK support [grant number KTP012139].

References

[1] Hochenbaum J, Vallis O and Kejariwal A. Automatic anomaly detection in the cloud via statistical learning. *arXive*. 2017; abs/1704.07706.
[2] Chandola V, Banerjee A and Kumar V. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 2009; 41(3): 1-58.
[3] Koosha G. and Osmar Z R. Time series contextual anomaly detection for detecting market manipulation in stock market. IEEE International Conference on Data Science and Advanced Analytics (DSAA) 2016; 1-10.
[4] Fox A J. Outliers in time series. *Journal of the Royal Statistical Society: Series B (Methodological)* 1972;
[5] Bates J M and Granger C W J. *The Combination of Forecasts*. OR 1969; 4(20): 451-468.
[6] Foorthuis R. On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics* 2021;12(4): 297-331.
[7] Nielsen A. Practical time series analysis: Prediction with statistics and machine learning. O'Reilly Media 2019.
[8] Song X, Wu M, Jermaine C and Ranka S. Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering* 2007; 19(5): 631-645.
[9] Hodge V J and Austin J. A survey of outlier detection methodologies. *Artificial intelligence review* 2004; 22(2): 85-126.
[10] Taylor S J and Letham B. Forecasting at scale. *The American Statistician* 2018; 72(1): 37-45.