ABSTRACT

Classical anomaly detection is principally concerned with point-based anomalies, anomalies that occur at a single data point. In this paper, we present a new mathematical model to express range-based anomalies, anomalies that occur over a range (or period) of time.

1 INTRODUCTION

Anomaly detection (AD) seeks to identify atypical events. Anomalies tend to be domain or problem specific, and many occur over a period of time. We refer to such events as range-based anomalies as they occur over a range (or period) of time. Therefore, it is critical that the accuracy measures for anomalies, and the systems detecting them, capture events that occur over a range of time. Unfortunately, classical metrics for anomaly detection were designed to handle only fixed-point anomalies [1]. An AD algorithm behaves much like a pattern recognition and binary classification algorithm; it recognizes certain patterns in its input and classifies them as either normal or anomalous. For this class of algorithms, Recall and Precision are widely used for evaluating the accuracy of the result. They are formally defined as in Equations 1 and 2, where TP is true positives, FP is false positives, and FN is false negatives.

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (1)
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (2)
\]

While useful for point-based anomalies, classical recall and precision suffer from their inability to capture, and bias, classification correctness for domain-specific time-series anomalies. Because of this, many time-series AD systems’ accuracy are being misrepresented, as point-based recall and precision are used to measure their effectiveness [8]. Furthermore, the need to accurately identify time-series anomalies is growing due to the explosion of streaming and real-time systems [2, 4, 7, 9]. To address this, we redefine recall and precision to encompass range-based anomalies. Unlike prior work [2, 6], our mathematical definitions are a superset of the classical definitions, enabling our system to subsume point-based anomalies. Moreover, our system is broadly generalizable, providing specialization functions to control a domain’s bias along a multi-dimensional axis that is necessary to accommodate the needs of specific domains.

In this short paper, we present our core technical contribution: novel formal definitions of recall and precision for range-based anomaly detection that both subsume those formerly defined for point-based anomaly detection as well as being customizable to a rich set of application domains. Empirical data has been omitted to meet the venue’s compressed format.

Table 1: Notation

| Notation | Description |
|----------|-------------|
| $R$      | set of real anomaly ranges |
| $R_i$    | the $i^{th}$ real anomaly range |
| $P$      | set of predicted anomaly ranges |
| $P_j$    | the $j^{th}$ predicted anomaly range |
| $N_r$    | number of real anomaly ranges |
| $N_p$    | number of predicted anomaly ranges |
| $\alpha$ | relative weight of existence reward |
| $\beta$  | relative weight of overlap reward |
| $\gamma$ | overlap cardinality function |
| $\delta$ | positional bias function |

2 RANGE-BASED RECALL

Classical Recall rewards an AD system when anomalies are successfully identified (i.e., TP) and penalizes it when they are not (i.e., FN). It is computed by counting the number of anomalous points successfully predicted and then dividing that number by the total number of anomalous points. However, it is not sensitive to domains where a single anomaly can be represented as a range of contiguous points. In this section, we propose a new way to compute recall for such range-based anomalies. Table 1 summarizes our notation.

Given a set of real anomaly ranges $R = \{R_1, \ldots, R_{N_r}\}$ and a set of predicted anomaly ranges $P = \{R_1, \ldots, R_{N_p}\}$, our $Recall(R, P)$ formulation iterates over the set of all real anomaly ranges ($R$), computing a recall score for each real anomaly range ($R_i$ $\in$ $R$) and adding them up into a total recall score. This total score is then divided by the total number of real anomalies ($N_r$) to obtain an average recall score for the whole time-series.

\[
Recall(R, P) = \frac{\sum_{i=1}^{N_r} Recall(R_i, P)}{N_r} \quad (3)
\]

When computing the recall score $Recall(R_i, P)$ for a single real anomaly range $R_i$, we take the following aspects into account:

- **Existence**: Identifying an anomaly (even by a single point in $R_i$) may be valuable in some application domains.
- **Size**: The larger the size of the correctly predicted portion of $R_i$, the higher the recall score will likely be.
- **Position**: In some cases, not only size, but also the relative position of the correctly predicted portion of $R_i$ may be important to the application (e.g., early and late biases).
- **Cardinality**: Detecting $R_i$ with a single predicted anomaly range $P_j$ $\in$ $P$ may be more valuable to an application than doing so with multiple different ranges in $P$.

We capture these aspects as a sum of two reward terms weighted by $\alpha$ and $\beta$, respectively, where $0 \leq \alpha, \beta \leq 1$ and $\alpha + \beta = 1$. $\alpha$ represents the relative importance of rewarding existence, whereas $\beta$ represents the relative importance of rewarding size, position, and
The most one predicted anomaly range (i.e., it is identified by a single in robotic defense), then the tail-end bias function should be used. If later index positions are more important (e.g., delayed response important than later ones (e.g., early cancer detection [5], real-time then the flat bias function should be used. If earlier ones are more important, which can be used with many different for positional bias as shown in Figure 1b. If all index positions are equally important, then the flat bias function should be used. If earlier ones are more important than later ones (e.g., early cancer detection [5], real-time apps [2]), then the front-end bias function should be used. Finally, if later index positions are more important (e.g., delayed response in robotic defense), then the tail-end bias function should be used.

Our recall formula for range-based anomalies subsumes the classical one for point-based anomalies (i.e., Recall$_T$ ≡ Recall) when:

(i) all $R_i \in R$ and $P_j \in P$ are represented as single-point ranges (e.g., range [1, 3] represented as $[1, 1, [2, 2], [3, 3]$), and
(ii) $\alpha = 0, \beta = 1, \gamma = 1, \omega(\cdot)$ as in Figure 1a, and $\delta(\cdot)$ returns flat positional bias as in Figure 1b.

3 RANGE-BASED PRECISION

Classical Precision is computed by counting the number of successful prediction points (i.e., TP) in proportion to the total number of prediction points (i.e., TP+FP). The key difference between Precision and Recall is that Precision penalizes FPs. In this section, we extend classical precision to handle range-based anomalies. Our formulation follows a similar structure as Recall$_T$.

Given a set of real anomaly ranges $R = \{R_1, ..., R_{N_p}\}$ and a set of predicted anomaly ranges $P = \{P_1, ..., P_{N_p}\}$, Precision$_T(R, P)$ iterates over the set of predicted anomaly ranges ($P$), computing a precision score for each range ($P_j \in P$) and then sums them. This sum is then divided by the total number of predicted anomalies ($N_p$), averaging the score for the whole time-series.

$$Precision_T(R, P) = \frac{\sum_{j=1}^{N_p} Precision_T(R, P_j)}{N_p}$$

When computing Precision$_T(R, P)$ for a single predicted anomaly range $P_j$, there is no need for an existence reward, because precision by definition emphasizes prediction quality, and existence by itself is too low a bar for judging the quality of a prediction. This removes the need for $\alpha$ and $\beta$ constants. Therefore:

$$Precision_T(R, P_j) = CardinalityFactor(P_j, R) \times \sum_{i=1}^{N_p} \omega(P_i, P_j \cap R_i, \delta)$$

$\gamma(\cdot), \omega(\cdot),$ and $\delta(\cdot)$ are customizable as before. Furthermore, Precision$_T$ ≡ Precision under the same settings as in Section 2 (except $\alpha$ and $\beta$ are not needed). Note that, while $\delta(\cdot)$ provides a potential knob for positional bias, we believe that in many domains a flat bias function will suffice for Precision$_T$, as an FP is typically considered uniformly bad wherever it appears in a prediction range.
REFERENCES

[1] C. C. Aggarwal. 2013. Outlier Analysis. Springer.
[2] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. 2017. Unsupervised Real-time Anomaly Detection for Streaming Data. *Neurocomputing* 262 (2017), 134–147.
[3] V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly Detection: A Survey. *ACM Computing Surveys* 41, 3 (2009), 15:1–15:58.
[4] S. Guha, N. Mishra, G. Roy, and O. Schrijvers. 2016. Robust Random Cut Forest Based Anomaly Detection on Streams. In *International Conference on Machine Learning (ICML)*. 2712–2721.
[5] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis. 2015. Machine Learning Applications in Cancer Prognosis and Prediction. *Computational and Structural Biotechnology Journal* 13 (2015), 8–17.
[6] A. Lavin and S. Ahmad. 2015. Evaluating Real-Time Anomaly Detection Algorithms - The Numenta Anomaly Benchmark. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*. 38–44.
[7] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. 89–94.
[8] N. Singh and C. Olinsky. 2017. Demystifying Numenta Anomaly Benchmark. In *International Joint Conference on Neural Networks (IJCNN)*. 1570–1577.
[9] Twitter. 2015. AnomalyDetection R Package. https://github.com/twitter/AnomalyDetection/. (2015).
[10] C. Warrender, S. Forrest, and B. Pearlmutter. 1999. Detecting Intrusions using System Calls: Alternative Data Models. In *IEEE Symposium on Security and Privacy*. 133–145.