Supplementary Material of CloudDet

I. Quantitative Evaluation

To evaluate the performance and scalability of the proposed anomaly detection algorithm, we test our proposed algorithm through a quantitative comparison with baseline methods based on Yahoo! S5 real time-series dataset [9].

A. Dataset, Baselines and Evaluation Metrics.

Dataset. In our evaluation, we use the A1Benchmark file of Yahoo! S5-A Labeled Anomaly Detection dataset, which contains 67 real time series with labeled anomalies. These real time series are collected from the real production traffic to some of the Yahoo! properties. Note that the timestamps of the A1Benchmark are replaced by integers with the increment of 1, where each data-point represents 1 hour worth of data.

Baseline Methods and its Parameter Settings. In order to decide which anomaly detection algorithms should be compared, two principles are applied: (1) cover typical anomaly detection techniques from different categories; and (2) the execution time is not much longer than our proposed algorithm. By surveying the existing paper, we choose five representative anomaly detection algorithms from five categories:

One-Class Support Vector Machine (oc-SVM), from classification-based algorithms, uses a hyperplane to distinguish two classes [4]. *RBF* (radial basis function) kernel is used in our system to deal with high-dimensional data. The kernel coefficient γ is chosen as the adjustable parameter for this algorithm when conducting accuracy evaluation, which can change proportionally from 0.05 to 0.95 with 0.1 as step length.

Local Outlier Factor (LOF) is also one of the neighbor-based analysis methods, but it is density-based [2]. It determines an outlier instance a by comparing a's k-neighborhood density to the k-neighborhood density of a's k-neighbors. We select k as the adjustable parameter for LOF's accuracy evaluation, which contains ten values growing proportionally from 4 to 40 with 4 as step length.

Robust Covariance Estimation (RCov) is a statistic-based algorithm that assumes the data follow a known distribution (e.g., Gaussian distribution). We use the Mahabolis distances to determine the outlyingness of a point from the known distribution. We set the proportion of points to be included in the support of the raw MCD (minimum covariance determinant) estimate, s, as the parameter for this algorithm's accuracy evaluation. It can change proportionally from 0.05 to 0.95 with 0.1 as step length.

Bitmap Detector (BD), based on symbolic aggregate approximation (SAX) of time series, is an assumption-free anomaly detection algorithm in time series with the bitmap [8]. We select the number of sections to categorize values as the adjustable parameter, which changes from proportionally from 2 to 20 with 2 as step length.

Twitter Anomaly Detection (Tad), also referred to as Seasonal Hybrid ESD (S-H-ESD), builds upon the Generalized ESD test for detecting anomalies. The algorithm employs piecewise approximation - this is rooted to the fact that trend extraction in the presence of anomalies in non-trivial - for anomaly detection. The periodic basis (the proportion of the time series length) is selected as the adjustable parameter, which changes from proportionally from 0.03 to 0.3 with 0.03 as step length.

We also tried Isolation Forest (iForest) [7] from model-based method, and the HOT-SAX [6] from SAX-based method for comparisons. However, we decided not to show their results due to their extremely long execution time compared with our proposed algorithms.

Evaluation Metrics. The evaluation metrics (ROC, execution time) are the standard information retrieval metrics used in many works related with time series anomaly detection [1], [5], [3]. We chose ROC for accuracy evaluation as the ratio between positive and negative instances is extremely imbalanced. We choose execution time to evaluate the scalability of our algorithm.

B. Implementations and Evaluation Settings

Implementations. We use the implementations of Rcov, LOF and SVM in the scikit-learn package [1], the implementation of BD in the luminol package[2], and the Tad implementation in the pyculiarity (A Python port of Twitter's AnomalyDetection R Package) [3]. All the parameters are set by default except for the adjustable parameter for each algorithm mentioned above.

Accuracy. First, we test the accuracy of each algorithm with ten different values for its chosen parameter (refer to Baseline Methods above). In particular, our proposed algorithm took the number of recent "historical data" L as the adjustable parameter (see Section 4.1 in paper), which growing from 5 to 50 with 5 as step length. Then we run each parameter setting of the algorithm ten times, with each time we choosing the proportion of anomalies as [0.005, 0.01, 0.02, 0.04, 0.08, 0.1, 0.3, 0.5, 0.7, 0.9] according to their anomaly score rankings. The first five fraction grow exponentially and the last five grow proportionally. Therefore, we got 100 runs (10 parameter values $\times$ 10 anomaly proportions) for each algorithm in accuracy evaluation, which reduces the bias of these algorithms to different datasets.

[1] https://scikit-learn.org/stable/modules/outlier_detection.html
[2] https://github.com/linkedin/luminol
[3] https://github.com/linkedin/luminol

To verify that our proposed algorithm is able to produce a good detection of time series anomalies, we benchmark the algorithm on a set of datasets selected from A1Benchmark. The datasets we use include: real_1, real_2 and real_3 as all the algorithms have a sound performance on these datasets. Specifically, real_1 has 1460 observations of which 13 are anomalies points. T

As shown in Fig., Overall, our algorithm (denoted as AS) outperformed the five baseline methods. In particular, our algorithm had a higher true positive rate when the false positive rate was low (below 0.2), which means that our algorithm can reduce false positive anomalies when detecting the same number of anomalies as the baseline algorithms. This is important for cloud computing anomaly detection because the experts can save effort in anomaly diagnosis when the data scale is very large.

Scalability. The algorithm proposed needs to be able to scale-out and efficient to cope with large-scale data generated by cloud computing system. To perform the evaluation, we tested the execution time of each algorithm by varying the length of the input time-series data. Specifically, in an experiment, the algorithm is performed on 50 time-series datasets from A1benchmark (real_1 to real_50), with each dataset running ten times due to their ten adjustable parameter values. We only vary the length of the datasets in different experiments to count the execution time of the algorithm. We vary the length of each dataset, from 100 to 700 points with 100 as step length, by selecting its first 100–700 data points. The experiment was conducted in an eight-core (Intel Core i7-4790 CPU@3.6GHz) Window 10 computer with 16 GB memory. The results are summarized in Fig. . The figure suggests that the execution time of our algorithm can scale and exhibits linearly with the length of the time series. "AS" is less scalable than RCov and LOF, probably due to the time spent for pattern matching before anomaly detection, but the difference is acceptable considering the higher accuracy in detection results compared with other algorithms.

C. Limitations.

Although the results show that our proposed algorithm has a sound performance when considering the speed and accuracy together, the validity of the results needs further evaluation. There exist some factors that may affect the validity, such as the selection of parameters and their settings, the bias in the tested datasets and the insufficiency of the evaluation metrics. In particular, our algorithm may perform worse than other baselines when the anomalies have specific pattern like short-term spikes. Therefore, the results of proposed algorithm can be affected by the aggregation strategy of anomaly scores from three components. A more optimal or automated aggregation method should be proposed in the future work.

REFERENCES

[1] B. Agrawal, T. Wiktorski, and C. Rong. Adaptive real-time anomaly detection in cloud infrastructures. *Concurrency and Computation: Practice and Experience*, 29(24):e4193, 2017.

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *ACM SIGMOD Record*, vol. 29, pp. 93–104. ACM, 2000.

[3] N. Cao, C. Lin, Q. Zhu, Y.-R. Lin, X. Teng, and X. Wen. Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):23–33, 2018.

[4] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[5] S. Huang, C. Fung, C. Liu, S. Zhang, G. Wei, Z. Luan, and D. Qian. Arena: Adaptive real-time update anomaly prediction in cloud systems. In *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–9. IEEE, 2017.

[6] E. Keogh, J. Lin, and A. Fu. Hot sax: Finding the most unusual time series subsequence: Algorithms and applications. In *Proc. of the 5th IEEE Int'l. Conf. on Data Mining*, pp. 440–449. Citeseer, 2004.

[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

[8] L. Wei, N. Kumar, V. N. Lolla, E. J. Keogh, S. Lonardi, and C. A. Ratanamahatana. Assumption-free anomaly detection in time series. In *SSDBM*, vol. 5, pp. 237–242, 2005.

[9] Yahoo. S5-a labeled anomaly detection dataset. "https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70".

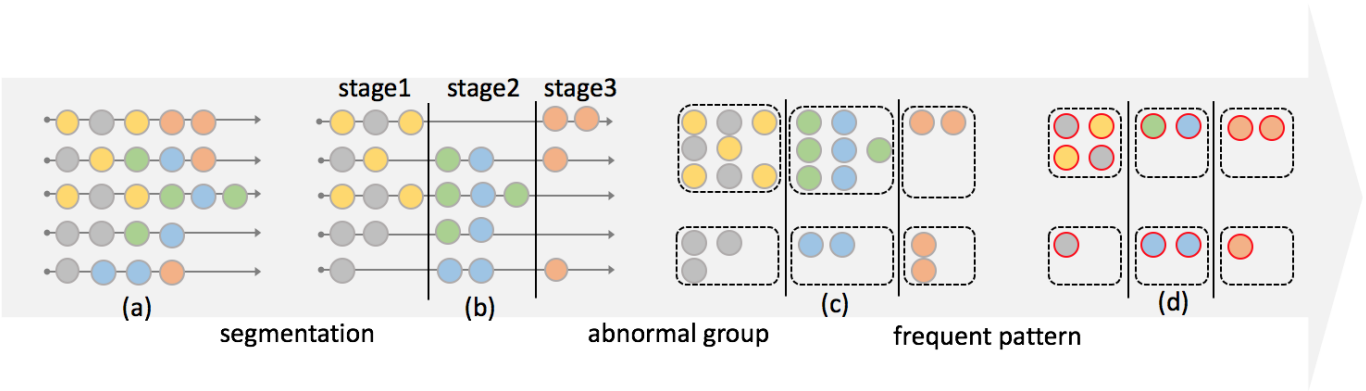

Fig. 1. The visual anomaly detection of learning sequence data includes three major step lengths: (1) sequence segmentation, (2) anomalous group identification, and (3) frequent pattern extraction.