Unveiling Data Deviations: Exploring Drift, Anomaly, and Outlier Identification Methods

Tajwar Mahmood
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Muhammad Hamza Nadeem
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Laiba Raza
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Umama Naseer
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Identifying and Visualizing data drift, anomalies and outliers in the era of fast-paced Data Analytics are essential to make your data legit and get the best performance out of your model. This paper is a comprehensive study and visualization of the occurrences through the Alibaba Cloud dataset, which is well known for its tremendous variety of data. Because the Alibaba Cloud dataset is vast and complex, it is essential to use proper methodologies in compiling and interpreting the data correctly. We then use these methods to detect outliers in CPU usage and several other metrics from the Alibaba Cloud dataset. We share an insight into the methods used to detect abnormal behavior in the Alibaba Cloud dataset.**

***Index Terms*—Drifts, Outliers**

I. Introduction

Anomaly detection is the process that evaluates datasets to identify data points that do not match the organization's standard data points [2]. Unusual anomaly detection is employed by companies to detect differences from those baselines and to evaluate conflicting data [4]. When manual monitoring is unrealistic, data mining and anomaly detection techniques help organizations monitor the vast amount of data within the company's IT infrastructure [5]. These techniques help security teams to identify abnormal data points that deviate from typical patterns, enabling real-time monitoring systems to prevent breaches, detect fraud and assess system health, which strengthens the company's security posture and reduces the risk of data exposure [1]. Anomaly detection effectiveness depends on the quality and training of the detection models. It is also critical for identifying true outliers. Anomaly detection becomes complicated with challenges such as poor data quality, insufficient training samples, and lack of pre-labeled data. Anomaly detection is effective in fraud prevention, cybersecurity, and network monitoring applications, allowing companies to boost their security by detecting and responding to irregular patterns [3].

II. Methodology

This research paper follows a structured methodology to analyze anomalies, outliers and data drifts within the CPU usage dataset of the Alibaba Cloud dataset using Python. The approach includes three key phases: data preparation, usage of Python-based techniques and results. Each phase is detailed below.

A. Tools and Frameworks

Python, a programming language, was used due to its data analysis, visualization, and anomaly detection libraries.

B. data preparation

To ensure quality and consistency, the CPU usage dataset of Alibaba Cloud was first preprocessed. This includes:

- Cleaning: Removing rows with missing or invalid values.
- Rolling Window Sizes: The dataset was further divided into rolling windows containing 10000 data points.

III. Result

A. Overview of CPU Usage Patterns

Figure 1 provides an in-depth analysis of CPU usage patterns across various window sizes. Importantly, it reveals the key insights into the system's performance and workload behavior over time. It should be noted that most histograms show a highly right-skewed distribution, which indicates that the lower the CPU usage values, higher the frequency. Subsequently, majority of tasks executed on the CPU are lightweight and require low CPU power. There is a noticeable peak at the lower end of the CPU usage spectrum, reaffirming that the majority of tasks do not demand substantial CPU resources. This illustrates that the system frequently handles tasks that do not require significant computational power. If we compare window sizes 50000 and 60000 with window sizes 80000 and 90000, the window sizes 50000 and 60000 have a sharper decline in frequency after the peak. This indicates specific periods have more sustained low CPU usage, while

others experience a more significant drop-off, possibly due to differences like tasks or system states.

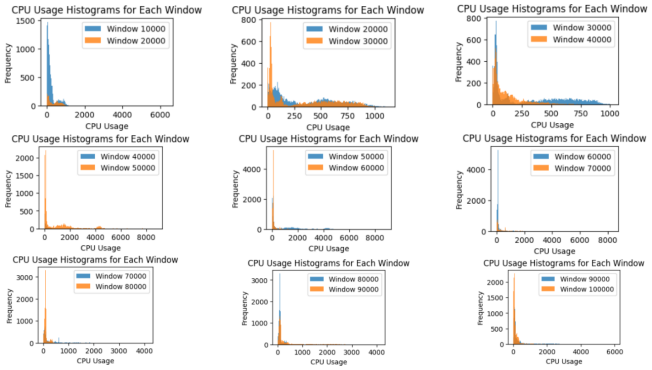

Fig. 1: Distribution Analysis of Window Size = 10000

B. Interpreting Histogram Patterns

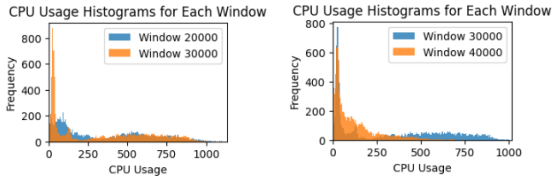

Fig. 2: Side-by-side comparison of CPU usage histograms.

Figure 2 describes two histograms each of which illustrates the relation of the data points in two separate time intervals. The left histogram illustrates the distribution of the values within the window 20,000 (light blue) and among the last window 30,000 (orange). To the right we have the histogram comparing window 30,000 (light blue) and window 40,000 (orange).

Kolmogorov-Smirnov statistic is the comparison of two distributions. A higher KS statistic mean that the measure of divergence is greater showing that the distribution tested is less similar.

In the left histogram KS statistic = 0.22 shows that window 20,000 and window 30,000 have small difference in distribution. This suggests that data within these two windows is somewhat comparable. In contrast, the right histogram shows larger distribution difference between window 30,000 and window 40,000 with KS statistic of 0.37. This means that the data within these two windows are out of phase and are dissimilar to the data of the first two windows.

Thus, analyzing the KS statistics and creating histograms that show how the actual and flagged data are similar over different Window sizes. Information about fluctuations is important to define changes within the data and choose the next steps based on the patterns received.

C. Line Plot

Figure 3 shows many drifts in CPU usage of Alibaba dataset. The x-axis represents individual data points of CPU

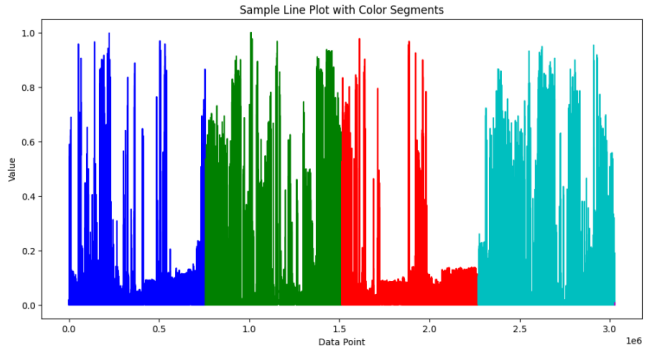

Fig. 3: Divided Data points in coloured Segments

usage, while the y-axis shows the normalized value of CPU usage (on a scale from 0 to 1).The sudden changes in CPU usages indicates that many CPU intensive tasks are executed at various times.

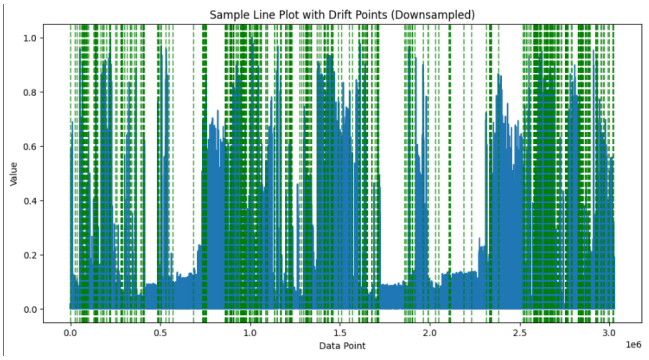

Fig. 4: Showing Drift Points using Line plot

The figure 4 is a clear demonstration of our observation that the CPU is running a high CPU utilization tasks at certain times which results in drifts . A drift is detected if the absolute difference between the maximum and minimum values in the window is greater than or equal to 5. This indicates a sudden, significant change in the class labels.

D. Box Plot

The Figure 5 represents Outliers in CPU usage data from the Alibaba Cloud dataset. The boxplot reveals a highly skewed distribution, with most data concentrated around low CPU usage, while the whiskers extend considerably, indicating significant variability. The numerous outliers suggest bursts of CPU activity. The long right whisker highlights the wide variability in CPU usage while CPU utilization is often low, there are periods of significant activity that need to be managed carefully.

E. Z method

Figure 6 shows the CPU usage values at certain points in time, where yellow points refer to the normal distribution, while red points refer to which were detected as outliers by the Z-score method. Of course, this leads to the fact that the

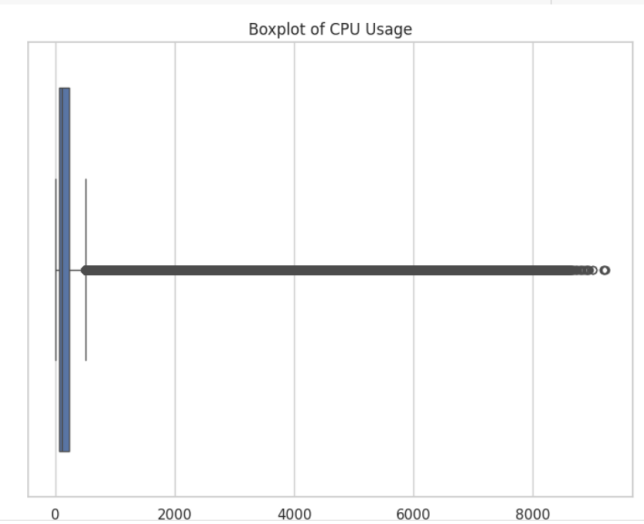

Fig. 5: show Outliers using Boxplot

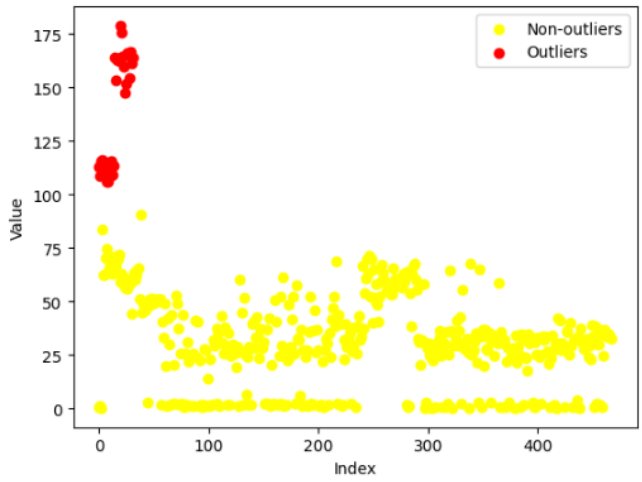

Fig. 6: Showing Outliers in CPU usage

red points are densely located towards the beginning of the index suggesting a probable shift of measured value of CPU usage by a great amount. The yellow dots scattered throughout the index range indicate the normal behavior of the Systems Thus, the values of an anomaly at index 0-50 are considerably higher suggesting that the system's behavior is different at this stage.

F. Scatter method

In Figure 7 , The x-axis represents data points from 0 to 2.5 million.The y-axis ranges from 0 to 1. The Blue points represents the general dataset values. The Red points represents drift points. Majority of data points are Blue which suggests the presence of repeating patterns or concentrated clusters of data points.The red points are sparsely present across the plot which shows non-repeating patterns or sudden changes in CPU usage.They may correspond to events like

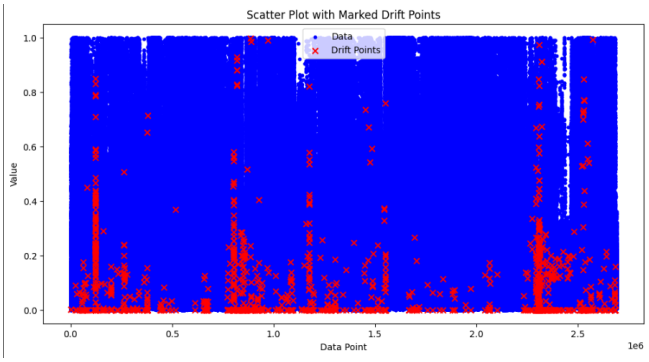

Fig. 7: Scatter Plot of CPU usage

changes in system behavior, shifts in user activity in Alibaba's operations.

IV. Conclusion

The findings of this study show that the Alibaba dataset has outliers and drift points which potentially indicates data quality issues and changes in data distribution over time. We have to implement effective strategies to mitigate the effects of outliers and drift using adaptive preprocessing, drift detection algorithms, and model retraining frameworks.

References

[1] Charu C Aggarwal. *Outlier analysis*. Springer, 2017.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[3] Jingkun Gao, Xiaomin Song, Qingsong Wen, Pichao Wang, Liang Sun, and Huan Xu. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *arXiv preprint arXiv:2002.09545*, 2020.

[4] Manoj Pareek, Sushil Gupta, Govinda Rajalu Lanke, and Dharmesh Dhabliya. Anamoly detection in very large scale system using big data. In *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*, 2023.

[5] Rui Ren, Jinheng Li, Lei Wang, Yan Yin, and Zheng Cao. Anomaly analysis and diagnosis for co-located datacenter workloads in the alibaba cluster. In *Benchmarking, Measuring, and Optimizing: Second BenchCouncil International Symposium, Bench 2019, Denver, CO, USA, November 14–16, 2019, Revised Selected Papers 2*, pages 278–291. Springer, 2020.