Statistical Detection of Collective Data Anomalies

Abstract

Attackers infiltrate databases and software systems to not only steal chunks of data and hold them for ransom, but also alter certain parts, setting up traps harder to discover and causing damages unlikely to recover. Without referring to data source, it is hard to detect a single record that is altered but still remain in correct value scopes. However, the change, especially to a subset of data, can be observed and detected in a collective scale. In this paper, we present a collective detection technique based on statistical divergence. Statistical divergence detection was explored in multimedia applications. But it is still underdeveloped in big data era due to data irregularity. The technique extracts distribution similarities among data collections, and then uses the statistical divergence to detect collective anomalies. Our technique continuously evaluates metrics as evolving features and calculate adaptive threshold to meet the best mathematical expectation. To illustrate details of the technique and explore its efficiency, we case-studied a real world problem of click farming detection against malicious online sellers. The evaluation shows that these techniques provided efficient classifiers. They were also sufficiently sensitive to a much smaller magnitude of data alteration, compared with real world malicious behaviours. Thus, it is applicable in the real world.

1 Introduction

Major improvements in data acquisition, transmission and storage are to increase data availability and affordability. Currently, there are more than 2.7ZB data in the digital universe [1] and the growing speed is doubling every two years. It has already been hard and will be much harder in the future to harness the exploding volume of data that has resulted in many problems in data management and engineering, threatening trustworthiness and reliability of data flows inside working systems. Data error rate in enterprises is approximately 1% to 5%, and for some, even above 30% [2]. Those data anomalies may arise due to both internal and external reasons.

On one hand, components inside systems may generate problematic source data. For example, in a sensor network, some sensors may generate erroneous data when it experiences power failure or other extreme conditions [3]. Data packages will be lost if sensor nodes fail to connect to network or some sensor hubs break down [4]. Also, human operators act as a heavily vulnerable part to bugs and mistakes. Malicious insiders even deliberately modify system configurations for fatal compromises [5]. A study shows that 65% of organizations state that human errors are the main cause of data problems [6] .

On the other hand, data manipulation [7] from outside hackers composes another potential threat of data quality and reliability. *Data Manipulation* here, according to a NSA definition, refers to that “hackers can infiltrate networks via any attack vector, find their way into databases and applications and change information contained in those systems, rather than stealing data and holding it for ransom”. If data is compromised, it will severely affect mining and learning algorithms and further change the final decision driven by the data. In 2013, hackers from Syria put up fake reports via Associated Press’ Twitter account and caused a 150-point drop in the Dow [8].

It is hard to detect a single record that is alerted but still remain in correct value scopes, but if sufficient data records are altered to change a final decision, we can still detect malicious data manipulation behaviours. According to our observation, typical manipulations on numerical data will lead to a drift or distortion of its original distribution. For measurable reshaping, we can enclose data collections with similar distribution patterns and filter out those strangely shaped ones. To address problems caused by data manipulation, we propose a novel technique which sorts out manipulated data collections from normal ones by adopting statistical divergence. Statistical divergence detection was explored in multimedia applications. But it is still underdeveloped in big data era due to data irregularity. In this paper, we focus on a concrete data manipulation problem: click farming in online shops, and try to apply our technique to pick out those dishonest sellers. Our techniques maps data collections to points in distribution spaces and reduce the problem to classical point anomaly detection. Optimizations estimate ground truth, mapping each data collection into a single real number within a definite interval. Then a Gaussian classifier can be applied to detect outliers derived from manipulated data. To automatically calculate adaptive threshold for the classifier, we keep two evidence sets for both normal points and anomalies, taking advantage of the property provided by statistical divergence. In the dynamic environments, these evidence sets are modified after every data collection is checked, in which manner they act intuitively as slide windows and keep up to the evolving features in dynamic scenarios. Our contribution includes: 1) A brief review on data anomaly detection and a study on the problem of click farming; 2) Detailed description of both basic and optimized framework of our technique, resolving several technical difficulties such as automated adaptive threshold; 3) Real world and synthetic data experiments that test efficiency of our technique and a comparison with a previous work on the same topic.

The rest of the paper is organised as follows: Section 2 states related work on data anomaly detection and describes a real world problem. Section 3 introduces statistical distance. Details of proposed technique are introduced in section 4. Then section 5 presents evaluation results and further findings of the algorithm. Finally, the paper is concluded in section 6.

2 Related Work

2.1 Data Anomaly Detection

Anomaly detection, also known as outlier detection, has been studied for a long time and discussed in diverse research domains, such as fraud detection, intrusion detection, system monitoring, fault detection and event detection in sensor networks. Anomaly detection algorithms deal with input data in the form of points (or records), sequences, graphs and spatial and geographical relationships. [9] According to relationships within data records, outliers can be classified into *point anomalies*, *contextual (or conditional) anomalies* and *collective anomalies*. [10]

Currently, distance based [11, 12] and feature evolving algorithms [13, 14, 15] catch most attention. Others adopted tree isolation [16], model based [17] and statistical methods [18] in certain applications.

To detect collective anomalies, [19] adopts the *ART (Adoptive Resonance Theory)* neural networks to detect time-series anomalies. *Box Modeling* is proposed in [20]. *Longest Common Subsequence* was leveraged in [21] as similarity metric for symbolic sequence. Markovian modeling techniques are also popular in this domain[22, 23, 24]. [25] depicts groups in social media as combinations of different "roles" and compare groups according to the proportion of each role within each group.

Wang et al. proposed a technique, *Multinomial Goodness-of-Fit* (MGoF), to analyze likelihood ratio of distributions via Kullback-Leibler divergence, and is fundamentally a hypothesis test on distributions [31]. MGoF divides the observed data sequence into several windows. It quantifies data in each window into a histogram and check these estimated distributions against several hypothesis. If the target distribution rejects all provided hypothesis, it is considered an anomaly and preserved as a new candidate of null hypothesis. If the target distribution failed to reject some hypothesis, then it is considered a supporting evidence of the one that yields most similarity. Furthermore, if the number of supporting evidence is larger than a threshold c_{th}, it is classified as non-anomaly.

MGoF is the best competitor out of the similar techniques, and we use it as our baseline against our approach.

2.2 Real World Problem: Click Farming Detection

Taobao possesses a market share of 50.6% to 56.2% in China by 2016 [26]. Currently, there are more than 9.4 million sellers in Taobao, providing more than 1 billion different products. Under the super-pressure caused by massive competitors, a number of the sellers choose to use some cheating techniques to raise reputation and sale volumes, then improve rankings in search lists.

The most popular approach to manipulate transaction and reputation data is *Click Farming*, where sellers use a large number of customer accounts to create fake transaction records and give high remarks on products. Professional click farmers are usually well organized groups or companies containing thousands of people. Some companies even develop professional applications that can be deployed on common PCs to improve productivity [27].

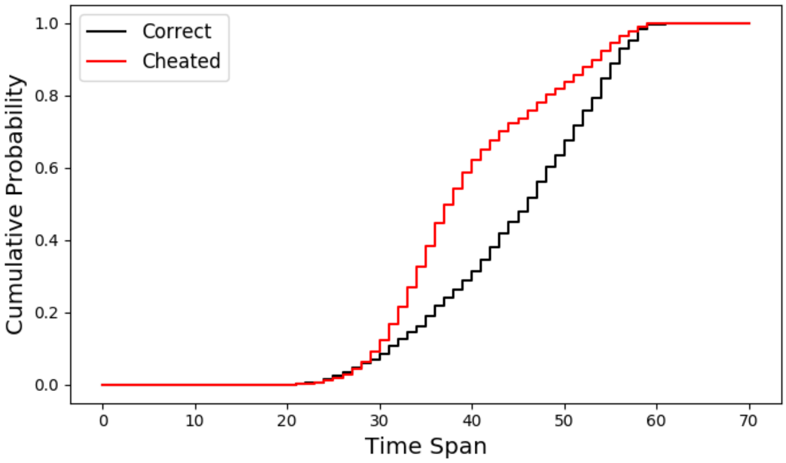

Figure 1: Example cumulative distribution function of original and click farmed daily transaction data

There are two types of click farming behaviours: Centralized and Equalized. Centralized click farming refers to the scenarios that transactions are randomly generated throughout the day. A significant feature of this approach is that the cheating transactions usually assemble together in a short period of time since most workers work at the same time. Equalized click farming refers to the circumstances that click farms are arranged by some well programmed applications or teams carefully managed and strictly commit transactions according to a timetable. Thus the transaction distribution may not vary too much with and without click farming.

A research performed in China showed that 81.9% of investigated people had heard of the behaviour of click farming, 51.2% are aware of click farm and 18.9% of them had experience of click farming themselves [28]. American researchers reported in 2015 that over 11000 sellers on Taobao were detected to have click farmed records and only 2.2% of 4000 investigated dishonest sellers had been penalized because of the cheating attempts [29].

Current detection techniques for click farming mainly focus on user behaviours, such as browsing frequencies and periods, most common purchasing time, favourite products, remarks and whether they communicate with sellers [30]. Those techniques require the platform to keep lots of records and user features. However, the detection can be easily bypassed by trained workers and some well programmed applications.

Although it is hard to classify users as honest or malicious, we can still find clues from the sellers' aspect. For normal sellers, their customers are usually similar since choices of products are seldom changed. Therefore, the distribution of transactions in a fixed period of time, say one day, is relatively stable. No matter how much alike between honest users and robots or the employed workers, the fake transaction records will always cause a bias or distortion of the original transaction distribution. To better observe the problem, we downloaded a real world data set containing Taobao online sellers' transaction records and emulated the circumstances if it had been click farmed (see section 5.1). Fig. 1 shows the difference between normal and click farmed distributions of one day in the data set. Thus, if we can measure the similarity between different transaction distributions, there is still a chance for us to detect dishonest sellers.

3 Preliminaries

Statistical divergence, also called statistical distance, measures the similarity between two or more distributions. Mathematically, statistical divergence is a function which describes the "distance" of one probability distribution to the other on a statistical manifold. Let $\mathbb{S}$ be a space of probability distributions, then a divergence is a function from $\mathbb{S}$ to non-negative real numbers:

$$D(\cdot||\cdot) : \mathbb{S} \times \mathbb{S} \to \mathbb{R}^+ \cup \{0\} \tag{1}$$

Divergence between two distributions P and Q, written as $D(P||Q)$, satisfies:

1. $D(P||Q) \geq 0, \forall P, Q \in \mathbb{S}$
2. $D(P||Q) = 0$, if and only if $P = Q$

For our purposes, we do not require the function D to have the property: $D(P||Q) = D(Q||P)$. But we do need it to be true that if Q is more similar with P than U, then $D(Q||P) < D(U||P)$. There are ways to calculate divergence, several frequently used divergence metrics are as follows:

3.1 Kullback-Leibler Divergence

Let P, Q be discrete probability distributions, $Q(x) = 0$ implies $P(x) = 0$ for $\forall x$, the *Kullback-Leibler Divergence* from Q to P is defined to be:

$$KLD(P||Q) = \sum_{Q(x)\neq 0} P(x) log\Big(\frac{P(x)}{Q(x)}\Big) \tag{2}$$

For P, Q being continuous distributions:

$$KLD(P||Q) = \int_{q(x)\neq 0} p(x) log \frac{p(x)}{q(x)} dx \tag{3}$$

3.2 Jensen-Shannon Divergence

Let P, Q be discrete probability distributions, *Jensen-Shannon Divergence* between P and Q is defined to be:

$$JSD(P||Q) = \frac{1}{2} KLD(P||M) + \frac{1}{2} KLD(Q||M) \tag{4}$$

where $M = \frac{1}{2}(P+Q)$.

A more generalized form is defined to be:

$$JSD(P_1, \ldots, P_n) = H\Big(\sum_{i=1}^{n} \pi_i P_i\Big) - \sum_{i=1}^{n} \pi_i H(P_i) \tag{5}$$

where H is Shannon Entropy, $M = \sum_{i=1}^{n} \pi_i P_i$ and $\sum_{i=1}^{n} \pi_i = 1$.

Especially, if $\pi_i = \frac{1}{n}$, then:

$$JSD(P_1, \ldots, P_n) = \frac{1}{n} \sum_{i=1}^{n} KLD(P_i || M) \tag{6}$$

Jensen-Shannon divergence has some fine properties:

1. $JSD(P||Q) = JSD(Q||P), \forall P, Q \in \mathbb{S}$.
2. $0 \leq JSD(P_1, \ldots, P_n) \leq log_k(n)$. If a k based algorithm is adopted.
3. To calculate $JSD(P||Q)$, it need not necessarily to be true that $Q(x) = 0$ implies $P(x) = 0$.

3.3 Bhattacharyya Distance

Let P, Q be discrete probability distributions over same domain X, *Bhattacharyya Distance* between P and Q is defined to be:

$$BD(P||Q) = -ln\Big(\sum_{x \in X} \sqrt{P(x)Q(x)}\Big) \tag{7}$$

3.4 Hellinger Distance

Let P, Q be discrete probability distributions, *Hellinger Distance* between P and Q is defined to be:

$$HD(P||Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{x} \left(\sqrt{P(x)} - \sqrt{Q(x)}\right)^2} \tag{8}$$

3.5 Kolmogorov-Smirnov Statistic

Let P, Q be discrete one-dimensional probability distributions, CDF_P and CDF_Q are their cumulative probability functions respectively, *Kolmogorov-Smirnov Statistic* between P and Q is defined to be:

$$KSS(P||Q) = \sup_{x} |CDF_P(x) - CDF_Q(x)| \tag{9}$$

Algorithm 1 SDD-R

```
Input: Data Collections $\mathbb{D} = \{D_1, \ldots, D_n\}$
Input: Estimated anomalous probability $\alpha$
Output: Anomalous Data Collections
for $i \leftarrow 1$ to $n$ do
    $P_i \leftarrow$ the distribution of $D_i$
end for
$P_R \leftarrow \frac{1}{n}\sum_{i=1}^{n} P_i$
for $i \leftarrow 1$ to $n$ do
    $d_i \leftarrow D(P_i||P_R)$
end for
$\mathcal{N}(\mu, \sigma) \leftarrow$ Gaussian distribution estimated by $d_i$
return $\{D_i | \frac{d_i - \mu}{\sigma} > 3\}$
```

4 Statistical Detection

Diverse data sets in the real world show certain structures caused by hidden patterns or relationships among records. For example, traffic volume in the highway and the business transaction records, they may show a relatively stable distribution in the daily scale. Manipulation on those data (e.g. Fig. 1) results in a drift or distortion of the distribution, which can be captured to trigger an alarm.

4.1 Statistical Divergence Detection with Reference(SDD-R)

From Section 3 we know that statistical divergence only provides a distance between two or more distributions. In a set of data collections, we can only draw a complete graph where nodes denote data collections and edges refer to the symmetric divergence between two connected nodes. From the graph we can find some points that have apparently larger distances with most of other points and return them as anomalies. This may work if anomalous nodes do not compose a large proportion. However the procedure will be too complicated to work out with large amounts of data. If it is assured that data collections form only one cluster, some optimizations can be applied to reduce complexity.

Alternatively we can provide a frame of reference that generates absolute coordinates rather than the relative ones. This optimization is feasible if data collections form one single cluster in distribution space. This is true in most reality scenarios given that distribution is adopted to depict a macro property which comes out as one universal conclusion. In other words , if multiple distributions are used to describe subgroups of entire sample space, then a conclusive one can be obtained by averaging all these sub-distributions. Therefore, we can use an estimate cluster center as reference and test distances between the reference and each other data collections(Algorithm 1), yielding absolute distances.

Fig. 2 shows distribution of all divergences against the reference. It can be approximated as a Gaussian distribution though the true one may differ a little more from

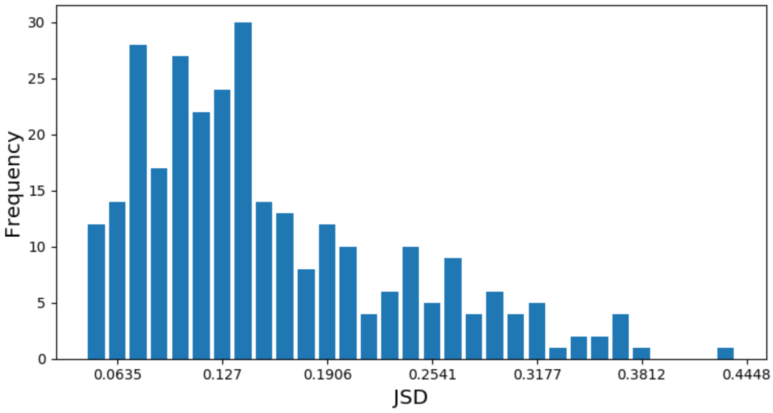

Figure 2: Distribution of Jensen-Shannon divergence on Taobao data set(without click farming) used in the experiments.

the standard Gaussian than the expected estimation error. That is due to the unknown randomness within real world data. Few assumptions can be applied in real world data sets, no mention that data volume is sometimes relatively low. This topic is out of the domain discussed in this paper and we here only introduce the technique instead of the specific distribution model. Certainly, if stronger assumptions can be included to provide a more precise model, this component in the framework can be replaced to give better results. For the simplicity of our proposal, we deem the distributions of divergences to be Gaussian.

By this approach, time complexity can be reduced from quadratic to linear. Fig. 8 in Section 5.2 demonstrates the result of the above process. Red distribution refer to the distances calculated from normal data collections, blue and green ones are from click-farmed data collections. Clearly, distances of normal data collections assembles together around a small value while anomalous ones lay around a larger distance value.

4.2 Optimization: Statistical Divergence Detection with Evidence(SDD-E)

It is possible to further optimize SDD-R if we can provide this algorithm with evidence(Algorithm 2).

Evidences enables the algorithm to not only refine estimation of real distribution but also build knowledge of anomalous collections, which is similar to the parameter estimation within a certain sample set.

According to the property of statistical divergence, we can infer that the true distribution of divergences calculated from normal data collections are close to but not exactly a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ since for each point, there are both definite upper and lower bounds instead of infinities. Therefore, μ should be slightly larger than zero($\mu = 0 \iff P_i = P_j, \forall P_i, P_j \in \mathbb{E}_N$, for real world data sets, this is highly

Algorithm 2 SDD-E

```
Input: Evidence set with normal data collections $\mathbb{E}_N = \{N_1, \ldots, N_n\}$
Input: Evidence set with anomalous data collections $\mathbb{E}_A = \{A_1, \ldots, A_m\}$
Input: Estimated anomalous probability $\alpha$
Input: New data collection $\mathbb{D} = \{D_1, \ldots, D_l\}$
Output: Anomalous data collections in $\mathbb{D}$
for $i \leftarrow 1$ to $n$ do
    $P_{N_i} \leftarrow$ distribution of $D_{N_i}$
end for
for $i \leftarrow 1$ to $m$ do
    $P_{A_i} \leftarrow$ distribution of $D_{A_i}$
end for
$P_R \leftarrow \frac{1}{n}\sum_{i=1}^{n} P_{N_i}$
for $i \leftarrow 1$ to $n$ do
    $d_{N_i} \leftarrow D(P_{N_i}||P_R)$
end for
for $i \leftarrow 1$ to $m$ do
    $d_{A_i} \leftarrow D(P_{A_i}||P_R)$
end for
$\mathcal{N}_N(\mu_N, \sigma_N) \leftarrow$ normal distribution estimated from $\{d_{N_1}, \ldots, d_{N_n}\}$
$\mathcal{N}_A(\mu_A, \sigma_A) \leftarrow$ normal distribution estimated from $\{d_{A_1}, \ldots, d_{A_m}\}$
$T \leftarrow$ proper threshold derived from $\mathcal{N}_N$, $\mathcal{N}_A$ and $\alpha$
for $i \leftarrow 1$ to $l$ do
    $P_i \leftarrow$ distribution of $D_i$
    $d_i \leftarrow D(P_i||P_R)$
end for
return $\{D_i | d_i > T\}$
```

unlikely). Time complexity for this algorithm is still linear but with a larger coefficient.

For certain divergence, it is possible to compare similarity from one distribution against multiple others, such as Jensen-Shannon Divergence. Although it reduces time complexity, it sacrifices unaffordable accuracy because divergence among multiple distribution dilutes differences. Take JSD as an example, suppose $P(1) = P(2) = P(3) = \frac{1}{3}$ and $Q(1) = \frac{1}{6}, Q(2) = \frac{1}{3}, Q(3) = \frac{1}{2}$, then $JSD(P||Q) \approx 0.033$ and $JSD(P, P, P, Q) \approx 0.024$.

This algorithm can be slightly modified to deal with concept drift(for example, trading trend changes over time for online shops as they are often in the process of expanding or dwindling) by turning the two evidence sets as sliding windows and adopting certain update strategies such as *Least Recently Used*(LRU). Time complexity for this optimization is $O(n \cdot (|\mathbb{E}_N| + |\mathbb{E}_A|) \cdot T_D)$, where T_D denotes time complexity of divergence calculation.

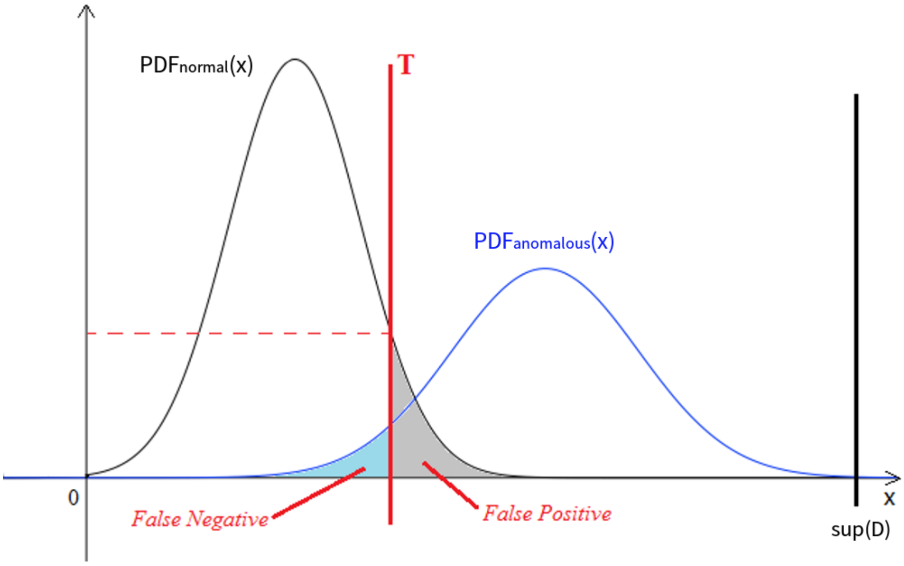

Figure 3: Threshold can be determined either by a probability density value, or a radius from the centre. In the scenario shown in the figure, the threshold can be determined by the position between two centres of the distribution, denoted as “T” here. This form of threshold can also be applied to other types of distributions even there is no intersection between these two.

4.3 Threshold

One important factor in algorithm SDD-E is the value of threshold. A lower threshold rejects more instances, improving the sensitivity of anomalous data while increasing the number of false alarms. A higher threshold provide higher true negative rates yet neglecting more possible threats.

A naive but prevalent approach is to set a fixed value as the threshold(As is shown in Algorithm 1). This approach is easy to implement and may give satisfying results in specific cases. However, a fixed threshold requires specific analysis in the certain scenario, manual observations and tuning of parameters, which involves lots of human labour. The rule of “3σ” declares all instances outside $[\mu - 3\sigma, \mu + 3\sigma]$ to be anomalous. It can be used to automatically determine a threshold. But as a rigid metric, it is merely an estimation of a suitable boundary considering average situations, which is far from optimality when concrete data is provided. It would be either lower than the optimum if anomalous data lies far away from the normal cluster, or higher than the optimum if the anomalies sit close to the cluster centre.

Fortunately, applying divergence as the distance measurement among data collections provides a fine property. With a reference distribution, divergences of normal data collections form a quasi-Gaussian distribution as we have seen in section 4.1. The same applies to those anomalous ones.

Moreover, as is verified in experiments, the meta-distribution of anomalous data collections lies in the right-hand-side to the normal one on the real line. As shown in Fig 3, the black curve ($PDF_{normal}(x)$ or in short $PDF_n(x)$) displays the probability density function (PDF) fitting those divergences calculated from normal data collections; the blue curve ($PDF_{anomalous}(x)$ or in short $PDF_a(x)$) displays the PDF derived from anomalous data collections. Threshold is chosen to minimize total errors(both false negative and false positive).

Suppose:

$$PDF_n(x) \approx \mathcal{N}(\mu_n, \sigma_n) \tag{10}$$

$$PDF_a(x) \approx \mathcal{N}(\mu_a, \sigma_a) \tag{11}$$

Then the optimal threshold T is calculated by E.q.(12). The optimal threshold will minimize total errors and yield an optimal outcome. However, this is not accurate enough, since E.q.(12) implicates an assumption that chances are the same for a new data collection to be either anomalous or not. If we can determine the probability for a new data collection to be anomalous in any segment of data sequence, the equation should be modified as E.q.(13), where α is the anomaly probability.

$$\begin{aligned}
T &= \arg\min_T \int_0^T PDF_a(x)dx + \int_T^{\sup(D)} PDF_n(x)dx \\
&\approx \arg\min_T \int_{-\infty}^T \frac{e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}}{\sqrt{2\pi}\sigma_a} dx + \int_T^{+\infty} \frac{e^{-\frac{(x-\mu_n)^2}{2\sigma_n^2}}}{\sqrt{2\pi}\sigma_n} dx \\
&= \begin{cases} \dfrac{1}{\sigma_a^2 - \sigma_n^2}\left[(\sigma_a^2\mu_n - \sigma_n^2\mu_a) \pm \sigma_a\sigma_n\sqrt{(\mu_a-\mu_n)^2 + 2(\sigma_a^2-\sigma_n^2)ln\dfrac{\sigma_a}{\sigma_n}}\right], & \sigma_a \neq \sigma_n \\ \dfrac{\mu_n+\mu_a}{2}, & \sigma_a = \sigma_n \end{cases}
\end{aligned} \tag{12}$$

Note: when $\sigma_a \neq \sigma_n$, keep the root s.t. $\dfrac{T-\mu_a}{\sigma_a^3}e^{-\frac{(T-\mu_a)^2}{2\sigma_a^2}} < \dfrac{T-\mu_n}{\sigma_n^3}e^{-\frac{(T-\mu_n)^2}{2\sigma_n^2}}$

$$\begin{aligned}
T &= \arg\min_T \alpha \int_0^T PDF_a(x)dx + (1-\alpha) \int_T^{\sup(D)} PDF_n(x)dx \\
&\approx \arg\min_T \alpha \int_{-\infty}^T \frac{e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}}{\sqrt{2\pi}\sigma_a} dx + (1-\alpha) \int_T^{+\infty} \frac{e^{-\frac{(x-\mu_n)^2}{2\sigma_n^2}}}{\sqrt{2\pi}\sigma_n} dx \\
&= \begin{cases} \dfrac{1}{\sigma_a^2 - \sigma_n^2}\left[(\sigma_a^2\mu_n - \sigma_n^2\mu_a) \pm \sigma_a\sigma_n\sqrt{(\mu_a-\mu_n)^2 + 2(\sigma_a^2-\sigma_n^2)ln\dfrac{(1-\alpha)\sigma_a}{\alpha\sigma_n}}\right], & \sigma_a \neq \sigma_n \\ \dfrac{\mu_n+\mu_a}{2} + \dfrac{k^2 ln\frac{1-\alpha}{\alpha}}{\mu_a-\mu_n}, & \sigma_a = \sigma_n = k \end{cases}
\end{aligned} \tag{13}$$

Note: when $\sigma_a \neq \sigma_n$, keep the root s.t. $\dfrac{\alpha(T-\mu_a)}{\sigma_a^3}e^{-\frac{(T-\mu_a)^2}{2\sigma_a^2}} < \dfrac{(1-\alpha)(T-\mu_n)}{\sigma_n^3}e^{-\frac{(T-\mu_n)^2}{2\sigma_n^2}}$

Moreover, with an estimated anomaly probability, SDD-R can be also optimized by ranking all data collections according to their divergence value and select first $n \cdot \alpha$ ones with highest values as anomalies.

5 Evaluation

Our algorithm was implemented and interpreted in Python 3.5.2. All experiments were tested on Ubuntu 16.04. In the following experiments, we figured out properties of real world data and performance of our technique against anomalous data collections. We also made a comparison among variations of SDD algorithms and MGoF.[1]

5.1 Methodology

We adopted two data sets: 1) Koubei sellers' transaction records[2]; 2) Synthetic random distribution data set. Koubei data set was provided by Alibaba Tian Chi big data competition where all records were collected from real world business scenarios. It contained information about seller features, user payments and browsing behaviour. We randomly chose one seller (ID: 1629) and extracted transaction history of this seller, records ranging from Nov. 11th 2015 to Oct. 31st 2016. Entire transaction set was then divided into 325 collections, each containing records in one day. Fig. 4 and 5 give an overview of it.

Two types of click-farmed data was generated according to patterns described in section 2.2. To emulate centralized click farming, we randomly inserted some Gaussian-distributed transactions in the chosen collection. As for emulating the equalized click farmers, we simply doubled each record in the chosen collection to make the new distribution exactly the same as the original one, which is harder for the online platform to discover. Usually, the click-farmed transactions are several times more than the volume it originally has, if the seller hires a group of organized workers. In our experiments, we use ν to denote the magnitude coefficient of click farming. Hence $|D_{anomalous}| = (1 + \nu)|D_{normal}|$. In the following experiments without extra illustration, we adopted $\nu = 1$.

One defect of this data set is that the detailed time stamp is aligned at each hour of the day due to desensitization. We constructed an enhanced data set by assigning every time stamp a random value for minutes and seconds. Therefore, the enhanced data set should be closer to the reality.

The synthetic data set was divided into four sections. First two sections contained sample sets drawn from a uniform and a Gaussian distribution respectively. The third section used a mixture of one uniform distribution and two Gaussian distributions to simulate a random-shaped distribution. Moreover, we made the random-shaped distribution drift slightly to form the last section of test data. Corresponding anomalies were drown from distributions with deviated parameters respectively.

[1] All resources and more detailed experiment results can be retrieved online: https://github.com/TramsWang/StatisticalAnomalyDetection

[2] https://tianchi.aliyun.com/competition/information.htm?raceId=231591

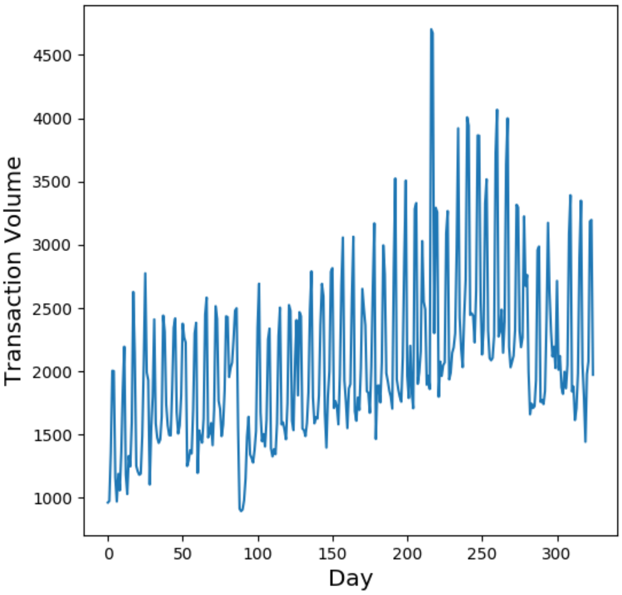

Figure 4: Changing of the daily sales volume shows that environment of online sales had been changing all the time.

We adopted histograms to depict distributions of any shape. Surely, the kernel density estimation approaches will give smooth and continuous estimations on any sampled data. But the computational cost will be too much to afford. Step size of histograms is chosen by:

$$l = c\sigma k^{-0.2}, \tag{14}$$

where k is sample size, c is a constant relative to the shape of distribution (e.g. for normal distribution, $c = 1.05$) and σ the standard deviation. For data sets with a large number of elements, a random sampling method, such as Monte-Carlo method, can be applied to speed up the estimation procedure.

Divergence metric adopted in each SDD algorithms was Jensen-Shannon divergence if no specific notation is made. However, MGoF used only Kullback-Leibler divergence due to its special mechanism. We use a "+" to denote algorithms optimized by a given α.

5.2 Experiments on Koubei Data Set

We first tested our algorithms on Koubei data set in order to see whether and why the algorithm works. Anomalies were random selected days replaced by corresponding click farmed version. To play the role of purchasing platform, we investigated two

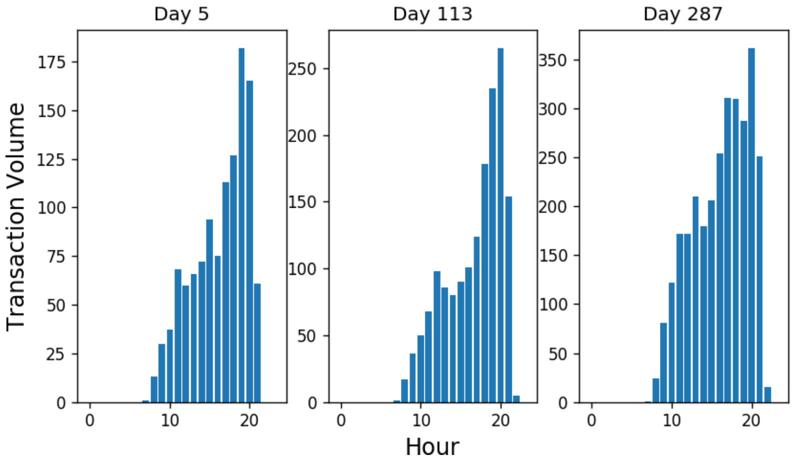

Figure 5: We selected 3 days randomly and drew sales distribution by counting hourly volume. Although sales volume has changed from day to day, the shape of the distribution remain almost alike.

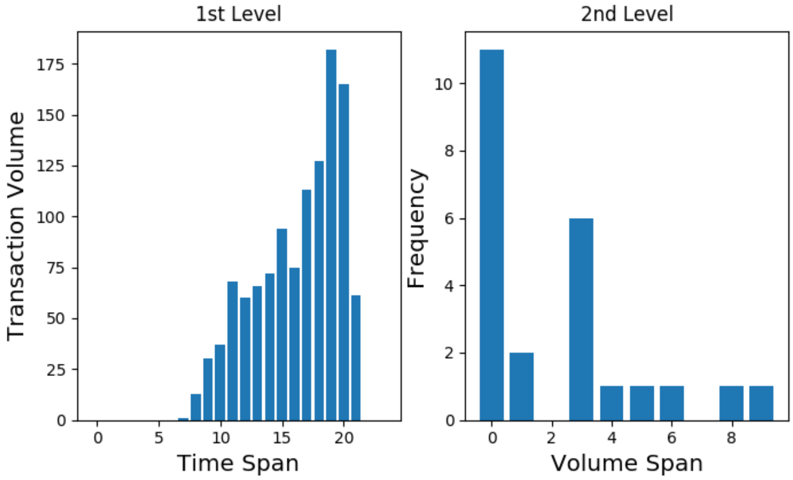

Figure 6: Examples of 1st and 2nd Level Histogram

levels of transaction distribution. The first level is to simply draw a histogram aligned to time spans. The second level is to draw a histogram on the sub-volumes in each time span(i.e. a histogram on frequencies in the first level histogram, as shown in Fig. 6).

On the raw data set, we had no choice but to set one hour a basket. While on the enhanced data set, we adopted E.q.(14) to determine step size automatically. To test SDD-E, we randomly selected 30 correct days and 10 click farmed days as normal and

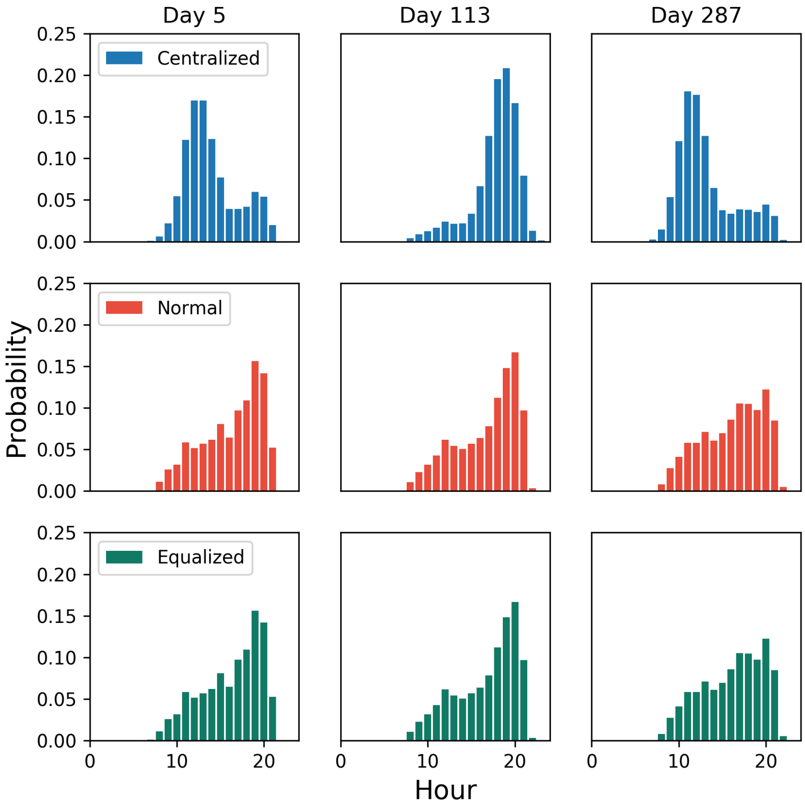

Figure 7: 1st level histogram of day 5, 113 and 287, each day in a column. Distributions after centralized and equalized click farming are in 1st and 3rd rows correspondingly. And the original distributions are shown in the 2nd row.

anomalous evidence respectively. Here $\alpha = 0.2$. The results are shown in Table 1 and 2.

When classifying toward 1st level histograms, centralized click farming behaviours can be easily discovered. As displayed in the first two rows in Fig. 7, normal collections share a similar distribution while centralized click-farmed ones abruptly violated the original shape. However, as a clever click farmer, equalized click farming did not in the least distort the distribution. Most of them escaped the check under perfect disguises. But when it came to 2nd level histograms, the “clever disguise” did not work any longer. It can be clearly seen in Fig. 8 that distribution of divergence of both click farming types shows an obvious deviation from the normal one.

The result showed that our technique outperformed MGoF in every real world

Table 1: Performance on Raw Data

	Centralized								**Equalized**							
	1st Level				**2nd Level**				**1st Level**				**2nd Level**			
	Pre(%)	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**
SDD-R	89.51	48.75	63.12	266.77	21.97	**99.38**	35.98	11.12	6.67	0.63	1.14	249.15	21.22	72.50	32.83	7.81
SDD-R+	91.25	91.25	**91.25**	**265.96**	**61.88**	61.88	61.88	10.08	9.38	9.38	9.38	**247.31**	**44.38**	44.38	44.38	6.92
SDD-E Static	**92.46**	68.75	78.86	292.50	36.55	98.13	53.26	5.75	6.67	0.63	1.14	271.45	36.07	86.25	50.86	5.64
SDD-E Static+	85.02	32.50	47.02	293.77	46.24	91.88	61.52	5.95	10.00	0.63	1.18	272.71	43.60	76.25	**55.48**	5.68
SDD-E Dynamic	49.11	**99.38**	65.73	699.97	23.01	**99.38**	37.37	245.65	10.36	**18.13**	**13.18**	681.09	22.09	**93.13**	35.71	242.85
SDD-E Dynamic+	73.21	98.75	84.09	701.06	48.02	96.25	**64.07**	255.43	8.15	6.88	7.46	681.89	40.79	78.13	53.59	253.03
MGoF	14.08	21.88	17.13	292.14	13.01	4.38	6.55	**3.64**	**12.50**	3.13	5.00	250.42	12.50	3.13	5.00	**3.71**

Table 2: Performance on Enhanced Data

	Centralized								**Equalized**							
	1st Level				**2nd Level**				**1st Level**				**2nd Level**			
	Pre(%)	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**
SDD-R	81.54	41.88	55.33	238.61	17.49	92.50	29.42	13.86	6.67	0.63	1.14	239.44	21.15	71.88	32.69	12.08
SDD-R+	**91.25**	91.25	**91.25**	**236.94**	36.88	36.88	36.88	12.98	8.13	8.13	8.13	**236.47**	38.75	38.75	38.75	11.03
SDD-E Static	88.31	67.50	76.52	257.60	31.99	92.50	47.54	9.74	6.67	0.63	1.14	257.65	34.49	91.25	50.06	9.37
SDD-E Static+	69.67	13.13	22.09	259.30	**42.12**	73.75	**53.62**	9.59	10.00	0.63	1.18	258.45	**44.17**	82.50	**57.54**	9.32
SDD-E Dynamic	42.93	**99.38**	59.96	1106.25	20.12	**95.63**	33.25	277.70	10.57	**17.50**	**13.18**	1108.93	20.30	**94.38**	33.42	271.74
SDD-E Dynamic+	69.18	98.13	81.15	1110.81	33.25	87.50	48.19	292.60	7.06	3.75	4.90	1118.00	37.21	80.63	50.92	283.50
MGoF	13.97	17.50	15.54	294.08	14.66	8.13	10.45	**8.01**	**12.50**	3.13	5.00	250.10	18.42	8.75	11.86	**7.55**

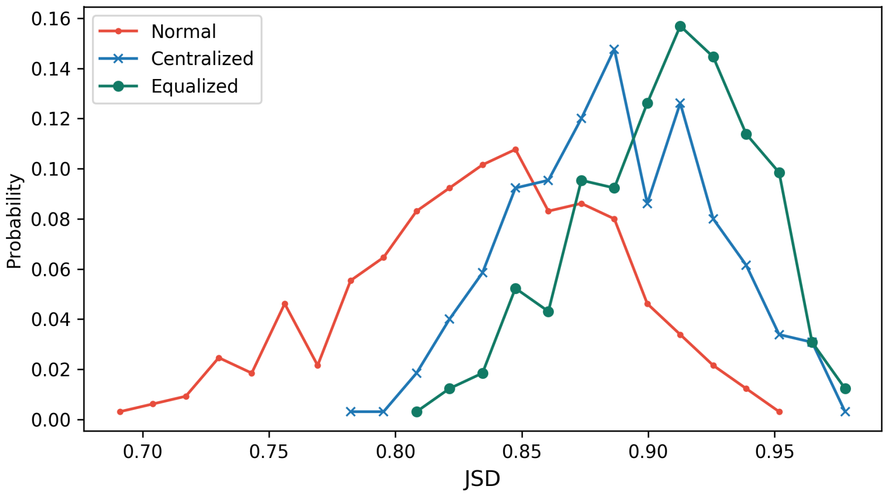

Figure 8: This figure shows distribution of JSD values(on 2nd level histograms) of normal and two types of click farming data. Divergences were calculated according to a reference averaged among all correct distributions.

cases. SDD-E provided best performance, yet it consumed the most computing power. Comparison among SDD-R revealed improvement of reference as well as the importance of threshold under this technique. Although dynamic SDD-E consumes more computation power, it is clear that dynamic SDD-E is capable of tracing the gradual shift of environment. MGoF turned out to be the worst since it always mark several false positive when c_{th} had not been met and much more false negatives when similar errors occurred too many.

Parameter α improved total accuracy of dynamic SDD-E algorithm by 10-20% as was supposed. It also increased its F1 by more than 20%. α made a great difference in SDD-R as well, which illustrated that divergence sorted almost all collections in correct order according to the averaged reference. However, static SDD-E did not show the same improvement. Since environment drift took greater influence in the result. In comparison with α, adaptive threshold given by evidence sets did not bring the most improvement. But this threshold can be applied together with other optimizations such as slide windows.

5.3 Test against Anomaly Proportion and Magnitude

In this experiment, we tested algorithm performance under various anomaly proportion and magnitude. α ranged from 0.1 to 0.9 when $\nu = 1$ and $\nu \in [0.1, 0.9]$ when $\alpha = 0.1$, other settings remains the same.

Fig. 9 shows that our technique outperformed MGoF and was relatively stable when dealing with all proportions of 1st level centralized anomalies. SDD-E performed even better since it maintains knowledge of both normal and anomalous distributions and

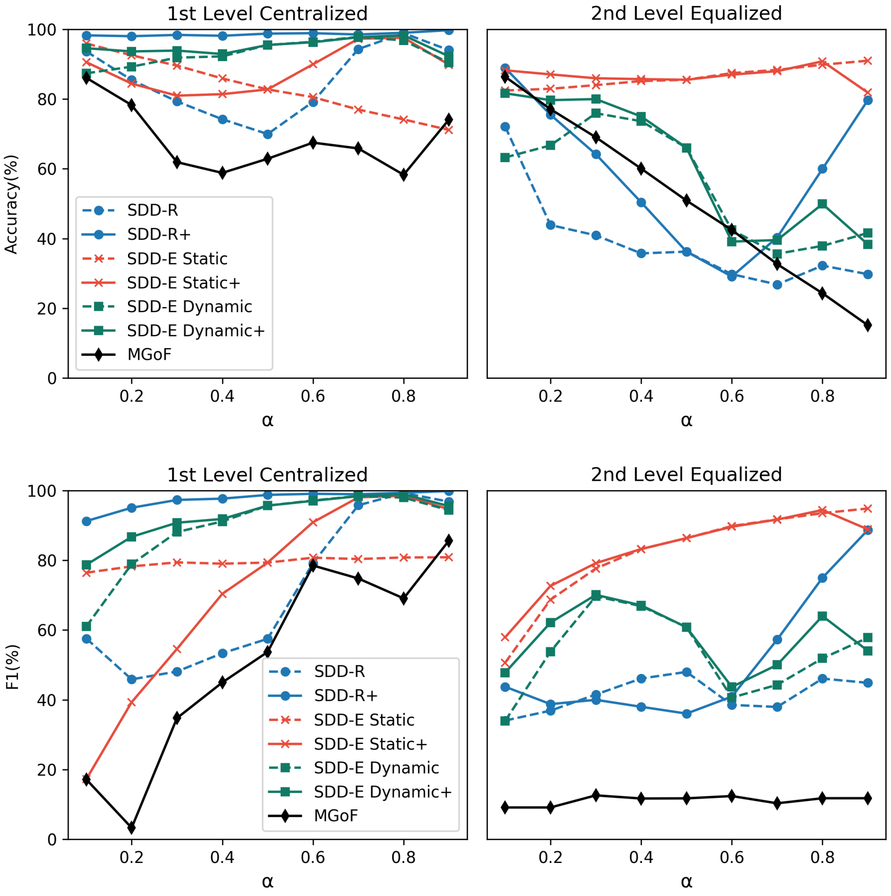

Figure 9: Accuracy and F1 on Different Anomaly Probabilities

calculates the threshold according to the best expectation. However, it relies on the accuracy of distribution estimation. When it came to 2nd level distributions, histograms became much coarser since data available was highly limited and thus its performance suffered dramatically.

MGoF tended to classify every distribution as anomaly, therefore benefited most by larger α. It always classifies as anomalous the first c_{th} distributions supporting every null hypothesis. Thus when α increased, the proportion of misclassified normal collections also became larger, while those anomalies were still considered anomalous. And given that the total number of normal collections drops down, the overall accuracy tended to increase as more instances are correctly classified as anomalous. However, the right half shows a different trend. One reason is that MGoF uses KLD other than JSD. In the Koubei dataset, the discrete estimation of distributions oscillated in a wide range, leading to that the prerequisite of KLD is often unsatisfied. Thus the calculation

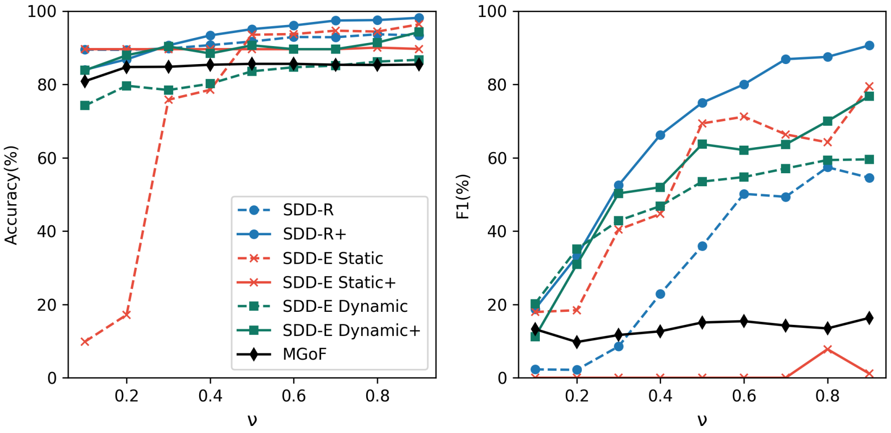

Figure 10: Accuracy and F1 on Different Anomaly Magnitudes

of KLD may not give a correct measurement. Furthermore, the 2nd histogram provided fewer probability entries than the 1st level did. Thus it shows a more significant deviation from our expectation. For the classifiers of MGoF, they compromised to a high error rate. Because more anomalies gathered together and the algorithm recognized them as clusters of normal data.

From Fig. 10 we can conclude that our algorithms are still the best, given that they are most sensitive toward tiny anomalous variations. However, static SDD-E did not rise until $\nu > 1$, this is because it suffered from fluctuation on the trade environment at the mean time. MGoF is not sensitive toward minor anomalies either. For a relatively small magnitude of click farming, the classifiers of MGoF quickly degrade to be trivial. The rigid threshold could not automatically rise up and was thus far from to optimal.

5.4 Results on Synthetic Data Set

In this experiment, we tested all seven algorithms on totally synthetic data sets. Results are shown in TABLE 3. It shows that our technique can be applied towards any kind of distributions. And these techniques worked better under irregular distributions since difference were clearer among these. Comparison between SDD-R and static SDD-E shows that adaptive thresholds provided more flexible classifiers. Results under random-shape drifting proves the efficiency of sliding windows toward drifting context.

The last experiment was carried out on random-shaped distribution data set, with $alpha = 0.1$ and rest parameters the same. Under different divergence metrics mentioned in section 3, F1 scores were calculated among all SDD algorithms and ROC curves were recorded on SDD-R and static SDD-E. Given that MGoF was defined specifically on Kullback-Leibler divergence, it cannot be tested in the same way. Results are shown in Fig. 11 and Fig. 12.

It is indicated that Jensen-Shannon divergence is suited to all techniques due to its

Table 3: Performance Comparison on Synthetic Data Sets

	Uniform				Gaussian				Random-shape				Random-shape with Drift			
	Pre(%)	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**	**Pre(%)**	**Rec(%)**	**F1(%)**	**T(ms)**
SDD-R	10.00	**100.00**	18.18	349.42	24.78	69.20	36.49	401.82	10.00	**100.00**	18.18	351.65	9.97	**99.00**	18.11	353.65
SDD-R+	22.40	22.40	22.40	348.49	34.40	34.40	34.40	**398.02**	58.40	58.40	58.40	**350.87**	1.20	1.20	1.20	**346.67**
SDD-E Static	43.71	82.60	57.17	372.74	33.34	66.60	44.44	410.72	69.24	93.20	79.45	372.77	12.34	97.20	21.91	369.63
SDD-E Static+	**75.10**	60.20	**66.83**	371.72	**45.93**	45.80	**45.86**	412.95	**89.52**	85.40	**87.41**	368.97	12.60	94.20	22.23	370.12
SDD-E Dynamic	18.53	81.80	30.21	6808.58	20.28	**71.40**	31.58	8985.92	25.90	97.40	40.92	5881.37	13.44	97.20	23.61	5747.19
SDD-E Dynamic+	27.85	25.60	26.68	8019.23	26.19	56.40	35.77	9197.94	77.60	79.40	78.49	6408.12	**50.55**	84.00	**63.11**	6176.57
MGoF	10.94	4.40	6.28	**347.97**	10.08	51.60	16.87	571.93	6.40	13.60	8.70	440.26	2.75	11.60	4.45	509.58

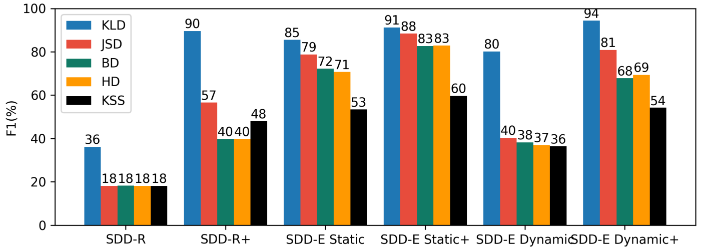

Figure 11: F1 under Different Divergence Metric

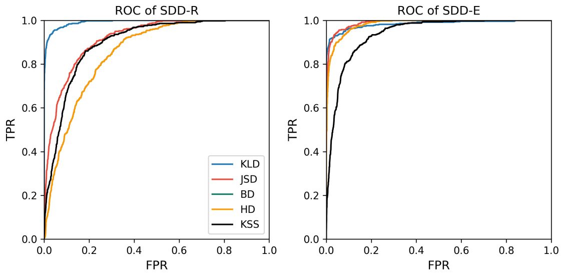

Figure 12: ROC under Different Divergence Metric

symmetry. Kullback-Leibler divergence provides more evident differences when references were given. Bhattacharyya distance and Hellinger distance turned out almost as good as Jensen-Shannon divergence, but they consumed less time. Kolmogorov-Smirnov Statistic performed relatively poor since it considers only the largest gap between two distributions, which provides little information.

5.5 Discussion

MGoF's learning procedure of anomalous probability hypothesis is inefficient. To maintain a comprehensive knowledge of anomalies, MGoF has to reserve a single hypothesis entry for every type of them. But in reality, it is always the case that we face the heterogeneity of outliers. In the Koubei data set, there can be tens of anomalous distributions caused solely by centralized click farming. It takes a long time to discover every possible type of anomaly. Besides, if there happens to be more than c_{th} anomalous distributions of the same type, later discovered collections will no longer declared to be anomalous any more.

However, in SDD-R and SDD-E, that is not a problem since it can map and gather all anomalies together and draw a universal boundary between them and all normal collections. These techniques are suitable to all typical divergence metrics and consumes little computation power(except dynamic SDD-E). The only drawback is they require comprehensive estimation of target distributions. Although other parameters need estimation as well, they are naturally addressable under big data circumstances.

6 Conclusion

This paper proposes a series of collective anomaly detection techniques, which helps detect data manipulations in modern data pipelines and data centres. Different from existing algorithms designed for collective anomalies, our approach employs statistical distance as the similarity measurement. We explored several technical points involved in the design of the algorithm and performed a thorough experiment to test its efficiency. The comparison experiment also illustrated the advantages of our technique. It can be concluded that the our technique can efficiently discover anomalies within the data collections and the classifier is sensitive enough toward real world data manipulations.

References

[1] (2017) Big data statistics and facts for 2017. [Online]. Available: https://www.waterfordtechnologies.com/big-data-interesting-facts/

[2] B. Saha and D. Srivastava, "Data quality: The other face of big data," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 1294–1297.

[3] M. A. Rassam, M. A. Maarof, and A. Zainal, “Adaptive and online data anomaly detection for wireless sensor systems,” *Knowledge-Based Systems*, vol. 60, pp. 44–57, 2014.

[4] H. Herodotou, B. Ding, S. Balakrishnan, G. Outhred, and P. Fitter, “Scalable near real-time failure localization of data center networks,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1689–1698.

[5] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, “Vc3: Trustworthy data analytics in the cloud using sgx,” in *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 2015, pp. 38–54.

[6] TowerData. (2013) 4 steps to eliminating human error in big data. [Online]. Available: http://www.towerdata.com/blog/bid/113787/4-Steps-to-Eliminating-Human-Error-in-Big-Data

[7] Is data manipulation the next step in cyber crime. [Online]. Available: https://www.cloudmask.com/blog/is-data-manipulation-the-next-step-in-cybercrime

[8] (2016). [Online]. Available: https://www.cnbc.com/2016/03/09/the-next-big-threat-in-hacking–data-sabotage.html

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[10] A. L. Goldberger *et al.*, “Components of a new research resource for complex physiologic signals, physiobank, physiotoolkit, and physionet, american heart association journals,” *Circulation*, vol. 101, no. 23, pp. 1–9, 2000.

[11] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner, “Scalable distance-based outlier detection over high-volume data streams,” in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 76–87.

[12] L. Cao, Y. Yan, C. Kuhlman, Q. Wang, E. A. Rundensteiner, and M. Eltabakh, “Multi-tactic distance-based outlier detection,” in *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 2017, pp. 959–970.

[13] M. M. Masud, Q. Chen, L. Khan, C. C. Aggarwal, J. Gao, J. Han, A. Srivastava, and N. C. Oza, “Classification and adaptive novel class detection of feature-evolving data streams,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1484–1497, 2013.

[14] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, “On the discovery of evolving truth,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 675–684.

[15] J. Shao, Z. Ahmadi, and S. Kramer, "Prototype-based learning on concept-drifting data streams," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 412–421.

[16] X. Zhang, W. Dou, Q. He, R. Zhou, C. Leckie, R. Kotagiri, and Z. Salcic, "Lshiforest: A generic framework for fast tree isolation based ensemble anomaly analysis," in *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 2017, pp. 983–994.

[17] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 625–636.

[18] Y. Zhu and D. Shasha, "Statstream: Statistical monitoring of thousands of data streams in real time," in *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 358–369.

[19] T. Caudell and D. Newman, "An adaptive resonance architecture to define normality and detect novelties in time series and databases," in *IEEE World Congress on Neural Networks, Portland, Oregon*, 1993, pp. 166–176.

[20] P. K. Chan and M. V. Mahoney, "Modeling multiple time series for anomaly detection," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 8–pp.

[21] S. Budalakoti, A. N. Srivastava, R. Akella, and E. Turkov, "Anomaly detection in large sets of high-dimensional symbol sequences," 2006.

[22] N. Ye *et al.*, "A markov chain model of temporal behavior for anomaly detection," in *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, vol. 166. West Point, NY, 2000, p. 169.

[23] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*. IEEE, 1999, pp. 133–145.

[24] D. Pavlov, "Sequence modeling with mixtures of conditional maximum entropy distributions," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 251–258.

[25] R. Yu, X. He, and Y. Liu, "Glad: group anomaly detection in social media analysis," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 2, p. 18, 2015.

[26] Iresearch. (2016). [Online]. Available: http://report.iresearch.cn/content/2016/11/265551.shtml

[27] M. Zhao, "On the phenomenon of click farming in taobao," *CHINA BUSINESS*, no. 12, pp. 31–33, 2016.

[28] Z. Yan, “Report of click farming phenomenon in taobao–protection for consumers’ right to know in online shopping,” *FaZhi Yu JingJi*, vol. 20, pp. 195–196, 2015.

[29] T. L. Mirror. (2016). [Online]. Available: http://tech.163.com/15/0405/14/AMENOJ7D00094ODV.html

[30] (2015). [Online]. Available: http://www.ebrun.com/20150906/147724.shtml

[31] C. Wang, K. Viswanathan, L. Choudur, V. Talwar, W. Satterfield, and K. Schwan, “Statistical techniques for online anomaly detection in data centers,” in *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. IEEE, 2011, pp. 385–392.