applied system innovation

MDPI

Article

IoT-Based Small Scale Anomaly Detection Using Dixon's Q Test for e-Health Data

Partha Pratim Ray * and Dinesh Dash

Department of Computer Applications, Sikkim University, Gangtok 737102, India; dd@nitp.ac.in
* Correspondence: parthapratimr.phd19.cs@nitp.ac.in

Abstract: Anomaly detection in the smart application domain can significantly improve the quality of data processing, especially when the size of a dataset is too small. Internet of Things (IoT) enables the development of numerous applications where sensor-data-aware anomalies can affect the decision making of the underlying system. In this paper, we propose a scheme: IoTDixon, which works on the Dixon's Q test to identify point anomalies from a simulated normally distributed dataset. The proposed technique involves Q statistics, Kolmogorov–Smirnov test, and partitioning of a given dataset into a specific data packet. The proposed techniques use Q-test to detect point anomalies. We find that value 76.37 is statistically significant where $P = 0.012 < \alpha = 0.05$, thus rejecting the null hypothesis for a test data packet. In other data packets, no such significance is observed; thus, no outlier is statistically detected. The proposed approach of IoTDixon can help to improve small-scale point anomaly detection for a small-size dataset as shown in the conducted experiments.

Keywords: IoT; anomaly detection; Dixon's Q test; small-size data packets; Kolmogorov–Smirnov test

check for updates

Citation: Ray, P.P.; Dash, D. IoT-Based Small Scale Anomaly Detection Using Dixon's Q Test for e-Health Data . *Appl. Syst. Innov.* **2021**, *4*, 100. https://doi.org/10.3390/asi4040100

Academic Editor: Giuseppe De Pietro

Received: 21 November 2021
Accepted: 13 December 2021
Published: 16 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

IoT has brought enormous opportunities to allow the developments of a multitude of smart applications, namely health monitoring, smart city, smart transportation, and smart industry. Sensors are used in IoT-based ecosystems to generate data streams in regular intervals to provide real-time monitoring support to the owner of the given IoT system [1]. Such sensors may sometimes become faulty or generate erroneous data which must be detected at an early stage; otherwise, it can create serious troubles for decision making in the following stage of the applications[2,3].

The problem becomes very difficult when the size of the dataset is too small. This can happen due to an abrupt change of one of the data point values compared to the rest [4–6]. Thus, the detection of outliers in a small-size dataset is a trivial task. This situation may be exaggerated when applied for an IoT-based system which is resource constrained in nature.

Performing high-end data analytics in a resource-limited IoT device is not always feasible. With existing deep learning and machine learning algorithms, one can find outliers from a dataset [7–9]. However, due to lack of hardware resources such as processor and memory, an IoT device may face severe difficulty [10]. Further, depicting a point anomaly from a very small-size dataset makes the whole process questionable [11,12].

In this paper, we propose the IoTDixon scheme to detect point anomalies from very small-size dataset for the IoT-based environment. The IoTDixon uses both Dixon's Q test and Kolmogorov–Smirnov test statistics to help find the anomaly. We consider a small-size dataset with 42 samples, which is assumed to be normally distributed. Such a dataset is further subdivided into six equal data packets, each with seven samples. We then perform the normality test of these data packets. Once this test is satisfied, the data packets are fed to Dixon's method for finding a point anomaly. Finally, we obtain an equal number of anomalies as of the data packets which are further investigated against the P values. If the P value is less than a given confidence level α, we infer it as statistically significant and declare the point as an outlier.

The key contributions of this work can be presented as follows:

- To propose IoTDixon scheme to detect point anomalies from small-size dataset;
- To integrate Dixon's Q test as a key statistic for detection of outlier points from small-size data packets;
- To integrate Kolmogorov–Smirnov test statistic as the normality checker.

Novelty of the work: Our work is the first ever study that uses Dixon's Q test and Kolmogorov–Smirnov test together to find small-size anomalies in IoT-based simulated scenarios. The study provides a scheme to divide a small dataset into further smaller data packets of constant size. We prefer to use the insertion sort to arrange the data points in ascending order for each of the data packets due to its faster response time. The presented IoTDixon algorithm has linear time complexity, thus making it appropriate for IoT-based devices.

The rest of the paper is presented as follows: Section 2 presents the formulation and derivation of Dixon's Q test. Section 3 presents the IoTDixon methodology. Section 4 provides results. Section 5 concludes the paper.

2. Dixon's Q Test

Dixon's Q test can be used to detect an anomaly from a dataset as follows [13–15]: We assume that a dataset contains n samples each denoted by x_i. Such samples must be arranged in ascending order as follows: $x_1 \leq x_2 \leq x_3 \leq x_4 \cdots \leq x_n$. We can define the statistic as Equation (1).

$$r_{j,i-1} = \frac{(x_n - x_{n-j})}{(x_n - x_i)} \tag{1}$$

The j on r is denoted as the number of anomalies which the data analyst suspects at the higher end of the given dataset. The i represents the number of anomalies that are suspected to be deposited at the lower end of the dataset.

The r values define six ways to perform different analytics based on the cumulative and density distribution functions, such as $r_{10}, r_{11}, r_{12}, r_{20}, r_{21}, r_{22}$ as shown below Equations (2)–(7) for $n \leq 30$ under the one-tail distribution. The r values are specified for following range of samples in the dataset $r_{10} : 3 \geq n \leq 7, r_{11} : 8 \geq n \leq 10$, $r_{21} : 11 \geq n \leq 13, r_{22} : 14 \geq n \leq 30$. However, it is slightly changed for a two-sided scenario as follows $r_{10} : 3 \geq n \leq 10, r_{11} : 8 \geq n \leq 10$, and $r_{21} : 11 \geq n \leq 13$.

$$r_{10} = \frac{(x_2 - x_1)}{(x_n - x_1)} \; OR \; \frac{(x_n - x_{n-1})}{(x_n - x_1)} \tag{2}$$

$$r_{11} = \frac{(x_2 - x_1)}{(x_{n-1} - x_1)} \; OR \; \frac{(x_n - x_{n-1})}{(x_n - x_2)} \tag{3}$$

$$r_{12} = \frac{(x_2 - x_1)}{(x_{n-2} - x_1)} \; OR \; \frac{(x_n - x_{n-1})}{(x_n - x_3)} \tag{4}$$

$$r_{20} = \frac{(x_3 - x_1)}{(x_n - x_1)} \; OR \; \frac{(x_n - x_{n-2})}{(x_n - x_1)} \tag{5}$$

$$r_{21} = \frac{(x_3 - x_1)}{(x_{n-1} - x_1)} \; OR \; \frac{(x_n - x_{n-2})}{(x_n - x_2)} \tag{6}$$

$$r_{22} = \frac{(x_3 - x_1)}{(x_{n-2} - x_1)} \; OR \; \frac{(x_n - x_{n-2})}{(x_n - x_3)} \tag{7}$$

2.1. Probability Density of r

The Dixon's ratio r follows Equation (1). However, joint probability density for x_i, x_n, and x_{n-j} can be obtained from Equation (8). We can use a combinatorial normalization

factor along with the density functions multiplied by the integration of possible values over the three variables excluding three points which are being used in the calculation. We can express the formulation based on the three observations such as $i-1$, $n-j-i-1$, and $j-1$ which are below x_j and within the range from x_i to x_{n-j} and from x_{n-j} to x_n, respectively. L and M are two variables, as shown in Equations (9) and (10).

$$P(x_i, x_{n-j}, x_n) = \frac{n!}{(i-1)!(n-j-i-1)!(j-1)!} * L * M \tag{8}$$

where,

$$L = [\int_{-\infty}^{x_i} \phi(t)dt]^{i-1} [\int_{x_i}^{n-j} \phi(t)dt]^{n-j-i-1} \tag{9}$$

and

$$M = [\int_{x_{n-j}}^{x_n} \phi(t)dt]^{j-1} \phi(x_i)\phi(x_{n-j})\phi(x_n) \tag{10}$$

We can use $\phi(t) = (2\pi)^{-\frac{1}{2}} exp[-\frac{x^2}{2}]$ as the density function of the given standard normal distribution. Equation (1) is obtained when $j = i = 1$, and we normally use r instead of t_{10} to avoid the ambiguity.

2.2. Jacobian Probability Density of r

The three variables x_i, x_{n-j}, and x_n can be expressed as $\{x, v, r\}$, where $v$denotes the Jacobian transformation. Now, the variables can be rearranged as follows: $x = x_n$, $v = x_n - x_i$,a dn $r = \frac{(x_n - x_{n-j})}{v}$. Now, to find the probability density of r on the Jacobian, we can integrate $-\infty < x < \infty$ and $0 \leq v < \infty$. Equation (11) shows the Jacobian evolved probability distribution of r. $\hat{L}$ and $\hat{M}$ present the variables as given in Equations (12) and (13), respectively.

$$P(r) = \frac{n!}{(i-1)!(n-j-i-1)!(j-1)!} * \hat{L} * \hat{M} \tag{11}$$

where,

$$\hat{L} = \int_{-\infty}^{\infty} \int_{0}^{\infty} [\int_{-\infty}^{x-v} \phi(t)dt]^{i-1} [\int_{x-v}^{x-rv} \phi(t)dt]^{n-j-i-1} \tag{12}$$

and

$$\hat{M} = [\int_{x-rv}^{x} \phi(t)dt]^{j-1} \phi(x-v)\phi(x-rv)\phi(x)v\, dv\, dx \tag{13}$$

2.2.1. Derivation of r_{10}

We can derive all the r with various i and j. We can find r_{10} when $j = i = 1$ as shown in Equation (14) and part calculations in Equations (15) and (16).

$$P(r_{10}) = \frac{n!}{(n-3)!} * \hat{L} * \hat{M} \tag{14}$$

where,

$$\hat{L} = \int_{-\infty}^{\infty} \int_{0}^{\infty} [\int_{x-v}^{x-rv} \phi(t)dt]^{n-3} \tag{15}$$

and

$$\hat{M} = \phi(x-v)\phi(x-rv)\phi(x)v\, dv\, dx \tag{16}$$

2.2.2. Derivation of r_{11}

We can find r_{11} when $j = 1$ and $i = 2$ as shown in Equation (17) and part calculations in Equations (18) and (19)

$$P(r_{11}) = \frac{n!}{(n-4)!} * \hat{L} * \hat{M} \tag{17}$$

where,

$$\hat{L} = \int_{-\infty}^{\infty} \int_{0}^{\infty} [\int_{-\infty}^{x-v} \phi(t)dt][\int_{x-v}^{x-rv} \phi(t)dt]^{n-4} \tag{18}$$

and

$$\hat{M} = \phi(x-v)\phi(x-rv)\phi(x)v\ dv\ dx \tag{19}$$

2.2.3. Derivation of r_{12}

We can find r_{12} when $j = 1$ and $i = 3$ as shown in Equation (20) and part calculations in Equations (21) and (22).

$$P(r_{12}) = \frac{n!}{2!(n-5)!} * \hat{L} * \hat{M} \tag{20}$$

where,

$$\hat{L} = \int_{-\infty}^{\infty} \int_{0}^{\infty} [\int_{-\infty}^{x-v} \phi(t)dt]^2[\int_{x-v}^{x-rv} \phi(t)dt]^{n-5} \tag{21}$$

and

$$\hat{M} = \phi(x-v)\phi(x-rv)\phi(x)v\ dv\ dx \tag{22}$$

2.2.4. Derivation of r_{20}

We can find r_{20} when $j = 2$ and $i = 1$ as shown in Equation (23) and part calculations in Equations (24) and (25).

$$P(r_{20}) = \frac{n!}{(n-4)!} * \hat{L} * \hat{M} \tag{23}$$

where,

$$\hat{L} = \int_{-\infty}^{\infty} \int_{0}^{\infty} [\int_{x-v}^{x-rv} \phi(t)dt]^{n-4} \tag{24}$$

and

$$\hat{M} = [\int_{x-rv}^{x} \phi(t)dt]\phi(x-v)\phi(x-rv)\phi(x)v\ dv\ dx \tag{25}$$

2.2.5. Derivation of r_{21}

We can find r_{21} when $j = 2$ and $i = 2$ as shown in Equation (26) and part calculations in Equations (27) and (28).

$$P(r_{21}) = \frac{n!}{(n-5)!} * \hat{L} * \hat{M} \tag{26}$$

where,

$$\hat{L} = \int_{-\infty}^{\infty} \int_{0}^{\infty} [\int_{-\infty}^{x-v} \phi(t)dt][\int_{x-v}^{x-rv} \phi(t)dt]^{n-5} \tag{27}$$

and

$$\hat{M} = [\int_{x-rv}^{x} \phi(t)dt]\phi(x-v)\phi(x-rv)\phi(x)v\ dv\ dx \tag{28}$$

2.2.6. Derivation of r_{22}

We can find r_{22} when $j = 2$ and $i = 3$ as shown in Equation (29) and part calculations in Equations (30) and (31).

$$P(r_{22}) = \frac{n!}{2!(n-6)!} * \hat{L} * \hat{M} \tag{29}$$

where,

$$\hat{L} = \int_{-\infty}^{\infty}\int_{0}^{\infty}[\int_{-\infty}^{x-v} \phi(t)dt]^2[\int_{x-v}^{x-rv} \phi(t)dt]^{n-6} \tag{30}$$

and

$$\hat{M} = [\int_{x-rv}^{x} \phi(t)dt]\phi(x-v)\phi(x-rv)\phi(x)v\ dv\ dx \tag{31}$$

2.3. Cumulative Distribution of R

we can rewrite Equations (8)–(10) in terms of cumulative normal distribution $\Phi(x)$ as shown in Equation (32)

$$P(r) = \frac{n!}{(i-1)!(n-j-i-1)!(j-1)!} * \overline{L} * \overline{M} \tag{32}$$

where,

$$\overline{L} = \int_{-\infty}^{\infty}\int_{0}^{\infty}[\Phi(x-v)]^{i-1} \tag{33}$$

and

$$\overline{M} = [\Phi(x-rv) - \Phi(x-v)]^{n-j-i-1}[\Phi(x) - \Phi(x-rv)]^{j-1}\phi(x-v)\phi(x-rv)\phi(x)v\ dv\ dx \tag{34}$$

The cumulative distribution function $CDF(R)$ can be expressed as Equation (35), where $0 \leq r \leq 1$ and $CDF(0) = 0$ and $CDF(1) = 1$. With a given probability α, we can find the roots as the critical values from Equation (36) with monotonic increment from 0 to 1.

$$CDF(R) = \int_{0}^{R} P(r)dr \tag{35}$$

$$CDF(R) = (1-\alpha) \tag{36}$$

2.4. Probability Density of r

Equation (32) can be rewritten on three ϕ terms with x^2, v^2 as shown in Equation (37) and part variables as shown in Equations (38) and (39). Herein, N represents the normalization factor with constant term $[2\pi]^{-\frac{3}{2}}$, and $J(x,r,v)$ depicts the terms with Φ.

$$P(r) = \frac{n!}{(i-1)!(n-j-1-1)!(j-1)!}[2\pi]^{-\frac{3}{2}}\int_{-\infty}^{\infty} e^{-\frac{3x^2}{2}}\int_{0}^{\infty} e^{-\frac{(1+r^2)v^2}{2}} * C * D \tag{37}$$

$P(r) = N\int_{-\infty}\infty e^{-\frac{3x^2}{2}}\int_{0}^{\infty} e^{-\frac{(1+r^2)v^2}{2}} J(x,v,r)e^{xv(1+r)}v\ dv\ dx$, where

$$C = [\Phi(x-v)]^{i-1}[\Phi(x-rv) - \Phi(x-v)]^{n-j-i-1}[\Phi(x) - \Phi(x-rv)]^{j-1} \tag{38}$$

$$D = e^{xv(1+r)} v \, dv \, dx \tag{39}$$

We can modify variable $t^2 = \frac{(1+r^2)v^2}{2}$ and $u^2 = \frac{3x^2}{2}$ to change the integration into the Gauss–Hermite quadrants as shown in Equation (40), where $x(u) = u\sqrt{\frac{2}{3}}$ and $v(t,r) = t\sqrt{\frac{2}{(1+r^2)}}$.

$$P(r) = N\sqrt{\frac{2}{3}}\sqrt{\frac{2}{(1+r^2)}} \int_{-\infty}^{\infty} e^{-u^2} \int_{0}^{\infty} e^{-t^2} J(x(u), v(t,r), r) e^{\frac{2ut(1+r)}{\sqrt{3(1+r^2)}}} dt \, du \tag{40}$$

Thus, the quadrature rules can be formulated as follows: Equations (41) and (42), where w_l, t_l represent weights and abscissas of the n_{hh} point belonging to half-range Hermite quadrature.

$$\int_{0}^{\infty} e^{-t^2} f(t) dt \approx \sum_{l=1}^{n_{hh}} w_l f(t_l) \tag{41}$$

$$\int_{-\infty}^{\infty} e^{-t^2} g(u) du \approx \sum_{k=1}^{n_{fh}} w_k g(u_k) \tag{42}$$

Now, the $CDF(R)$ can be computed as Equation (43), where w_m, y_m refer to the Gauss–Legendre weights and abscissas on the given range of $[-1, 1]$ with $y = \frac{2r}{R} - 1$ as the variable transformation on the range $[0, R]$ to $[-1, 1]$.

$$CDF(R) \approx \frac{R}{2} \sum_{m=1}^{n_g l} w_m P(\frac{R_{y_m}}{2}) \tag{43}$$

2.5. Range Test

The data analyst must use the Dixon's Q test once on any dataset. The Q test is performed by Equation (44). The gap denotes an absolute difference $|x - y|$, where x, y are real numbers. The metric properties on such absolute difference should hold the following inequalities, (i) $|x - y| \geq 0$, (ii) $|x - y| = 0$ when $x = y$, (iii) $|x - y| = |y - x|$, and (iv) $|x - z| \leq |x - y| + |y - z|$.

$$Q = \frac{Gap}{Range} \tag{44}$$

The Q is then tested against the $Q_{critical}$, i.e., table-wise reference value based on a given confidence interval and provided number of observations. A rejection is provided when $Q > Q_{critical}$; otherwise, an acceptance is made.

3. System Design

The IoTDixon flow chart is shown in Figure 1. The flow chart shows the process behind the proposed methodology where a small test data stream can be fed for anomaly detection. Initially, the small test dataset was divided into m data packets each with seven samples. We then performed each of the data packets for a test of normal distribution by using the Kolmogorov–Smirnov test. Upon notification as the normal distribution, the respective data packet was then processed for the Dixon's test. Finally, all the anomalies from each of the data packets were collected by the data analysts for further investigations.

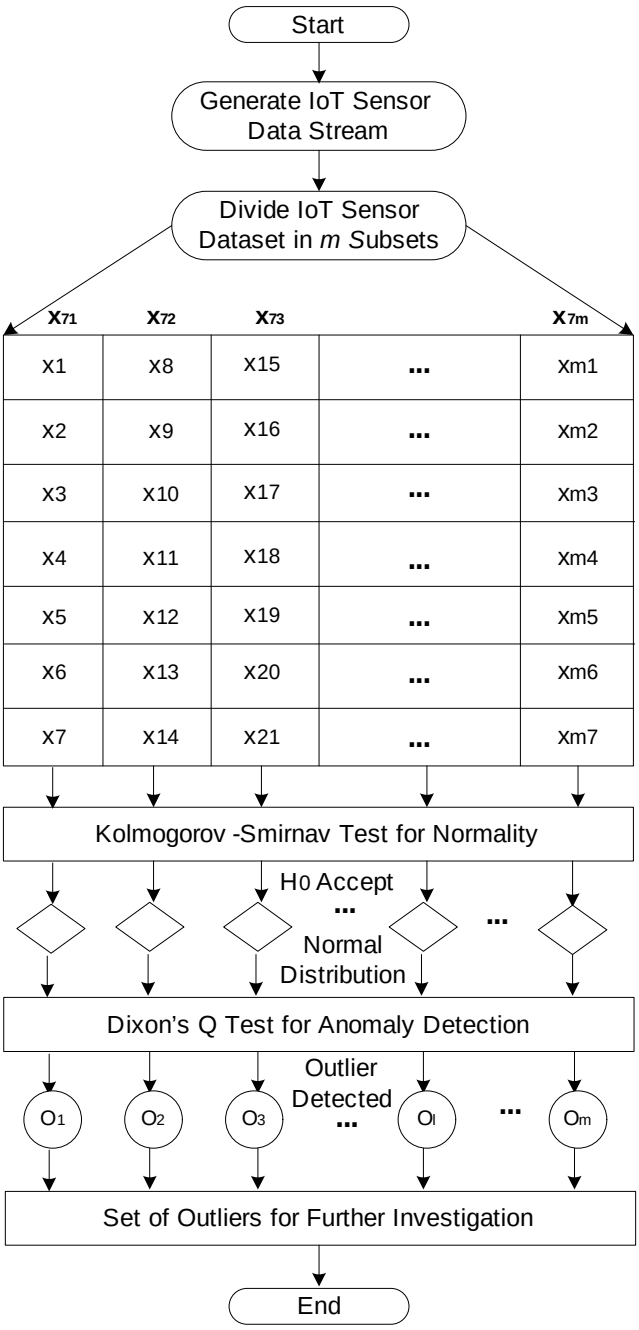

Figure 1. IoTDixon flow chart.

3.1. IoTDixon Algorithm

We present the IoTDixon algorithm for detection of single anomaly from a packet of samples taken from an small test data stream $X = \{x_1, x_2, x_3, \cdots, x_n\}$. The IoTDixon technique works as follows: We assume that IoT-based health sensor data are being streamed on a regular interval to an edge device. We assume that small test data stream is divided into $m = \lceil \frac{len(X)}{\eta} \rceil$ number of packets, where η is a given as the amount of samples in each data packet. We select the appropriate r statistic based on the sample size η. By default, we use r_{10} as the Dixon statistic where a sample of data packet must lie within the range of $[3, 7]$. However, one can change the range and r statistic according to the need to adjust with the packet size. The partitioning task required $\mathcal{O}(m)$ that depends on the η. We then perform the insertion sort on the j^{th} generated data packet $x_{\eta j}$. In this study, we select $\eta = 7$; however, as mentioned earlier, it can be changed to other values, though keeping

the minimum limit to three samples. We select insertion sort due to its faster response time for small datasets typically less than 10. Thus, we expect the sorting time to be around $\mathcal{O}(1)$, i.e., negligible. Then, we perform the normality checking procedure by using the Kolmogorov–Smirnov test in $\mathcal{O}(\eta)$ time. Once the given data packet is statistically proven to be normally distributed, then the data packet is forwarded for the two-sided Dixon's Q test. Finally, the outlier $o_{\eta j}$ from the sample data packet is returned. Thus, we can find the maximum m number of outliers as the total number of data packets is m. It is the duty of the data analyst to further investigate the obtained outliers to find the most effective anomaly. The overall time complexity of the proposed algorithm can be obtained in a linear time $\mathcal{O}(m+\eta)$. The IoTDixon Algorithm is shown in Algorithm 1.

Algorithm 1: IoTDixon Algorithm

```
Input: IoT-based small test data stream X where 1 ≤ i ≤ n and select appropriate
  r_{j,i-1} statistic for Dixon's Q test depending on sample packet size η
Output: Possible anomalies
Make packet x_{ηj} of each η amount of small test data samples
while (m=⌈len(X)/η⌉ > 0) do
    Perform insertion sort on the j^{th} data packet x_{ηj}
    Evaluate for normality by the Kolmogorov–Smirnov test where α = 0.05
    if ((x_{ηj})==Normally Distributed) then
        Perform two-sided Dixon's Q test on x_{ηj}
        Return an outlier o_{ηj} from the sample data packet
```

3.2. Kolmogorov–Smirnov Algorithm

The Kolmogorov–Smirnov (KS) test is used to evaluate whether a random sample selected from a dataset is drawn from a fixed normal distribution function $F(x)$, i.e., one-sample test [16,17]. It can also be used to evaluate whether two datasets belong to the same fixed distribution, i.e., two-sample test. The KS test requires no *a-priori* knowledge about the distribution of samples under consideration. In this study, we use a one-sample KS test on the data packet $x_{\eta j}$. We find $D^{+}_{max} = \sqrt{\eta}\ max\{\frac{t}{\eta} - F(x_t)\}, \forall t, 1 \leq t \leq \eta$, $D^{-}_{max} = \sqrt{\eta}\ max\{F(x_t) - \frac{t-1}{\eta}\}, \forall t, 1 \leq t \leq \eta$, and $D_{max} = max\{D^{+}_{max}, D^{-}_{max}\}$, where D^{+}_{max} represents the maximum positive), D^{-}_{max} refers to the maximum negative, and D_{max} denotes the maximum absolute. The null hypothesis H_0 is expressed as the data packet $x_{\eta j}$ follows the normal distribution. The alternative hypothesis H_1 is that the data do not follow the normal distribution. The H_0 can be rejected when the $D_{max} > D_{c\alpha}$ at the confidence level $\alpha = 0.05$. This infers that the data packet $x_{\eta j}$ is not normally distributed. Otherwise, we fail to reject the H_0 and infer that $x_{\eta j}$ is normally distributed. The Kolmogorov–Smirnov Algorithm is shown in Algorithm 2.

3.3. Dixon's Q Algorithm

We present the Dixon's Q algorithm where we provide the normally distributed small test data packet $x_{\eta j}$ which is ranked or ordered. A potential outlier sample x_t can be tested as follows: $Q = r_{10} = \frac{|(x_t - x_{t+1})|}{|(x_{max} - x_{min})|}$. If $Q < Q_{c\alpha}$, we fail to reject the null hypothesis H_0, which implies that the sample x_t is not an outlier. Otherwise, we infer to accept the null hypothesis H_0, which implies that the sample x_t is an outlier. In both cases, the null hypothesis H_0 can be stated as follows: there is no significant difference between the suspected data and the rest of $x_{\eta j}$; thus, it is not an outlier. Dixon's Q two–sided algorithm is shown in Algorithm 3.

Algorithm 2: Kolmogorov–Smirnov Algorithm

Input: IoT data packet $x_{\eta j}$
Output: Kolmogorov–Smirnov estimate D_{max}
$D^{+}_{max} = \sqrt{\eta}\ max\{\frac{t}{\eta} - F(x_t)\}, \forall t, 1 \leq t \leq \eta$
$D^{-}_{max} = \sqrt{\eta}\ max\{F(x_t) - \frac{t-1}{\eta}\}, \forall t, 1 \leq t \leq \eta$
$D_{max} = max\{D^{+}_{max}, D^{-}_{max}\}$
if $(D_{max} > D_{c\alpha=0.05})$ **then**
    Reject the null hypothesis H_0, implying that the $x_{\eta j}$ data packet is not normally distributed
else
    Fail to reject the null hypothesis H_0, implying that the data packet $x_{\eta j}$ is normally distributed

Algorithm 3: Dixon's Q two-sided Algorithm

Input: Normally distributed small test data packet $x_{\eta j}$ Output: A point outlier
Calculate Q statistic $Q = r_{10} = \frac{|(x_t - x_{t+1})|}{|(x_{max} - x_{min})|}$
if $(Q > Q_{c\alpha})$ **then**
    Reject the null hypothesis H_0, which implies that the sample x_t is an outlier
else
    Fail to reject the null hypothesis H_0, which implies that the the sample x_t is not an outlier

3.4. IoTDixon Dataset

We performed the study under the R distribution framework where we used the *dixonTest* package for performing the Q test. We created a dataset that has 42 samples with a mean of 72 and standard deviation of 2 to simulate the pulse rate per minute of a human being. The dataset is then partitioned into six equally sized subset of data packets having $\eta = 7$ named $x1, x2, x3, x4, x5, x6$. Such partitioned subsets are then used for checking anomalies per packet level.

4. Results

We obtained the IoTDixon normal distribution and evaluate them against (i) histogram with density curve (top left), (ii) plot of data points (top right), (iii) box plot (bottom left), and (iv) QQ plot (bottom right) for each of the six data packets. Figure 2 shows the packet wise evaluation of normality for following (a) x1 dataset, (b) x2 dataset, (c) x3 dataset, (d) x4 dataset, (e) x5 dataset, and (f) x6 dataset. We also present the overall normality evaluation for the whole dataset comprising 42 samples as in Figure 3.

We perform KS-test on each of the data packets. The cumulative distribution plots for each of the six data packets are shown in Figure 4, where x1, x2, x3, x4, x5, and x6 datasets are considered separately. All the six data packets are considered as normally distributed for proceeding the Dixon's Q test.

We perform the Dixon test on each of the 6 packets: x1, x2, x3, x4, x5, and x6. We find the Q statistic value for each of them as shown in Table 1. The probability of the Dixon test is shown as P. The position of selected anomaly from each data packet is mentioned under the POS column, and the corresponding value of anomaly point is shown under the anomaly column.

We find that value 76.37 is statistically significant where $P = 0.012 < \alpha = 0.05$, thus rejecting the null hypothesis for x6 data packet. In other data packets, no such significance is observed; thus, no outlier is statistically detected.

The proposed work is performed for the first time to showcase the use of the Dixon's Q-test for detecting point anomalies from a small-scale dataset. In this study, we used a

health dataset as a case study to validate the applicability of the proposed methodology. This approach can be deployed at a resource-constrained IoT-edge device connected to a health sensor as a proof of concept for the purpose of detection of point anomalies from a small-scale dataset. IoT devices are less processing capacity aware; thus, such systems should be fed with a lightweight scheme with a minimal amount dataset at their vicinity. Doing so can certainly improve the existing scenario of large-scale dataset-aware anomaly detection schemes toward minimalistic processing consumption deployments. Thus, the proposed technique can support at the edge real-life implications where small numbers of samples are collected and mitigated for localized anomaly detection. This can further minimize the overhead of a high amount anomaly eradication procedure in the later phase of application.

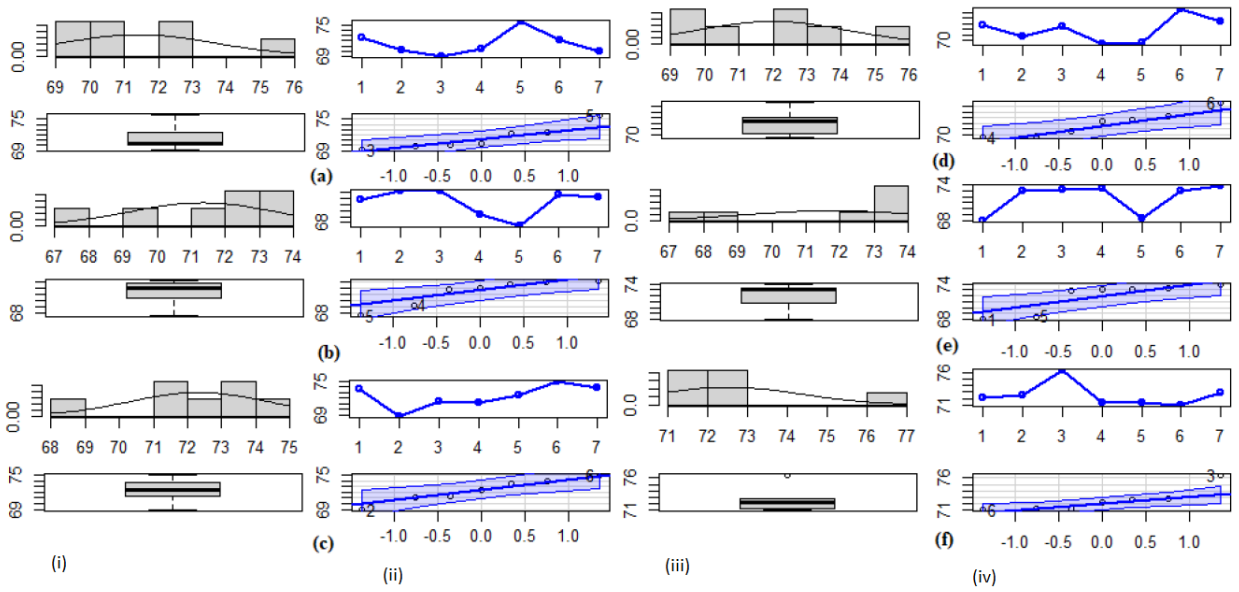

Figure 2. IoTDixon normal distribution evaluation (i) histogram with density curve (top-left), (ii) plot of data points, (iii) box plot, and (iv) QQ plot. (**a**) x1 dataset, (**b**) x2 dataset, (**c**) x3 dataset, (**d**) x4 dataset, (**e**) x5 dataset, and (**f**) x6 dataset.

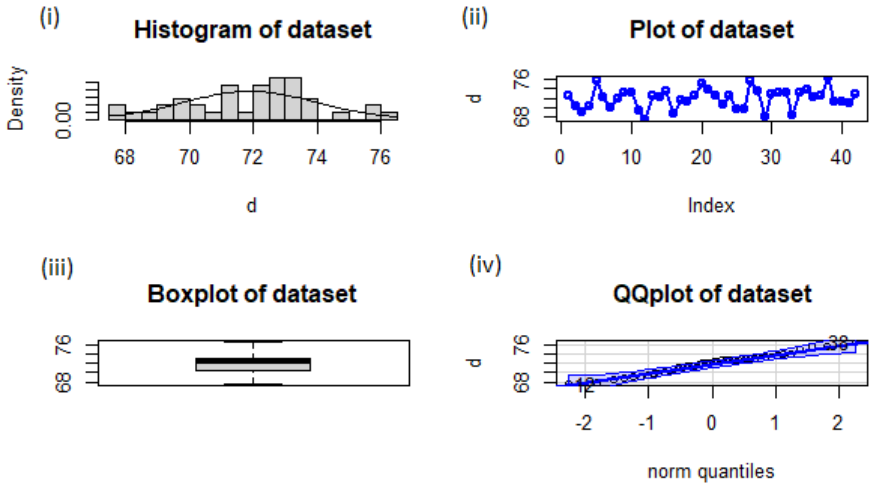

Figure 3. IoTDixon complete dataset normal distribution evaluation (**i**) histogram with density curve (top-left), (**ii**) plot of data points, (**iii**) box plot, and (**iv**) QQ plot.

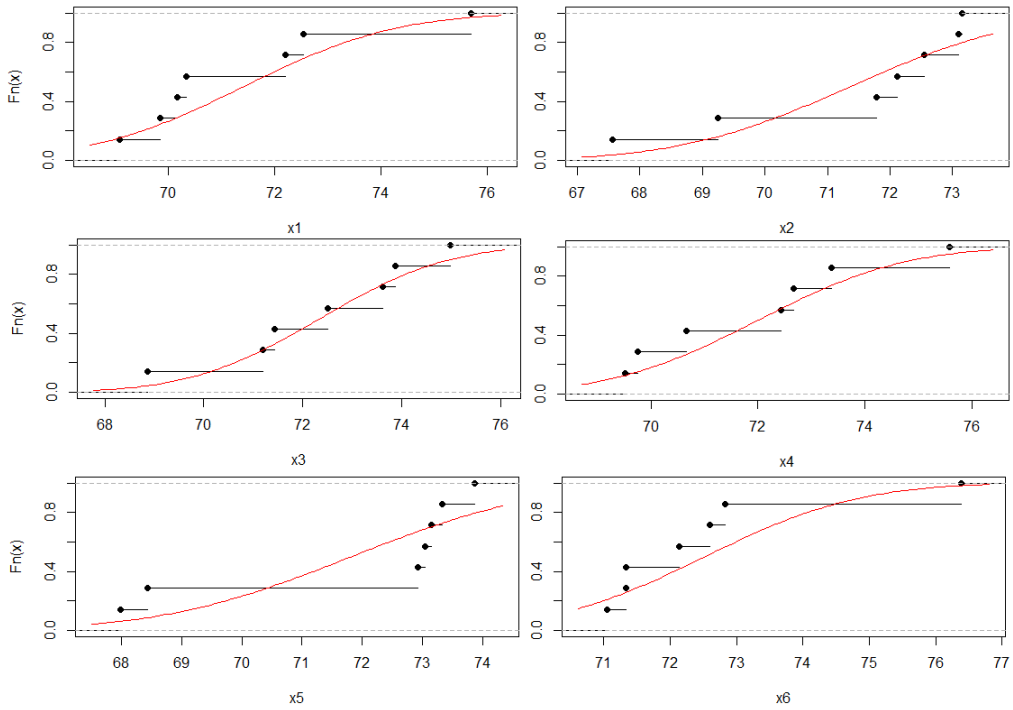

Figure 4. IoTDixon cumulative distribution function of x1, x2, x3, x4, x5, and x6 dataset (from top-left row-wise).

Table 1. IoTDixon aware anomaly detection.

	Q	P	POS	Anomaly
x1	0.477	0.1343	5	75.69
x2	0.300	0.533	5	67.56
x3	0.378	0.311	2	68.86
x4	0.365	0.344	6	75.58
x5	0.090	1	7	73.85
x6	0.665	0.012 ***	3	76.37 ***

5. Conclusions

This paper presents a novel IoTDixon methodology that can work on small-size data packets obtained from the given IoT dataset. As the Q test only provides a single anomaly from a small data packet, it can be useful for sensor data gathering wherein a few repetitions of averaging are performed. The proposed techniques uses Q-test to detect point anomalies. We find that value 76.37 is statistically significant where $P = 0.012 < \alpha = 0.05$, thus rejecting the null hypothesis for a test data packet. The IoTDixon algorithm can be useful in real-life applications with a small fragment of data analytics scenario, for example gathering small number of health data by an IoT sensor and identifying any anomaly present therein. Thus, anomaly detection can be imposed at the IoT-edge devices to detect possible point anomalies from a small set of data instead of searching them from a very large dataset that can be difficult in terms of power and processing capacity utilization by the resource-constrained IoT devices.

Author Contributions: P.P.R. conceptualized, investigated and written the paper, D.D. supported with expert advice in this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hansen, E.B.; Bøgh, S. Artificial intelligence and internet of things in small and medium-sized enterprises: A survey. *J. Manuf. Syst.* **2021**, *58*, 362–372. [CrossRef]
2. Han, G.; Tu, J.; Liu, L.; Martinez-Garcia, M.; Choi, C. An Intelligent Signal Processing Data Denoising Method for Control Systems Protection in the Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2021**, doi:10.1109/TII.2021.3096970. [CrossRef]
3. Haji, S.H.; Ameen, S.Y. Attack and anomaly detection in iot networks using machine learning techniques: A review. *Asian J. Res. Comput. Sci.* **2021**, 30–46. doi:10.9734/ajrcos/2021/v9i230218. [CrossRef]
4. Chen, Z.; Chen, D.; Zhang, X.; Yuan, Z.; Cheng, X. Learning Graph Structures with Transformer for Multivariate Time Series Anomaly Detection in IoT. *IEEE Internet Things J.* **2021**. doi:10.1109/JIOT.2021.3100509. [CrossRef]
5. Bhatia, M.P.S.; Sangwan, S.R. Soft computing for anomaly detection and prediction to mitigate IoT-based real-time abuse. *Pers. Ubiquitous Comput.* **2021**, 1–11. doi: 10.1007/s00779-021-01567-8. [CrossRef]
6. Fan, Z.; Feng, H.; Jiang, J.; Zhao, C.; Jiang, N.; Wang, W.; Zeng, F. Monte Carlo Optimization for Sliding Window Size in Dixon Quality Control of Environmental Monitoring Time Series Data. *Appl. Sci.* **2020**, *10*, 1876. [CrossRef]
7. Cauteruccio, F.; Cinelli, L.; Corradini, E.; Terracina, G.; Ursino, D.; Virgili, L.; Savaglio, C.; Liotta Al, F.G. A framework for anomaly detection and classification in Multiple IoT scenarios. *Future Gener. Comput. Syst.* **2021**, *114*, 322–335. [CrossRef]
8. Kayan, H.; Majib, Y.; Alsafery, W.; Barhamgi, M.; Perera, C. AnoML-IoT: An end to end re-configurable multi-protocol anomaly detection pipeline for Internet of Things. *Internet Things* **2021**, *16*, 100437. [CrossRef]
9. Yahyaoui, A.; Abdellatif, T.; Yangui, S.; Attia, R. READ-IoT: Reliable Event and Anomaly Detection Framework for the Internet of Things. *IEEE Access* **2021**, *9*, 24168–24186. [CrossRef]
10. Vangipuram, R.; Gunupudi, R.K.; Puligadda, V.K.; Vinjamuri, J. A machine learning approach for imputation and anomaly detection in IoT environment. *Expert Syst.* **2020**, *37*, e12556. [CrossRef]
11. Huang, K.; Chen, Z.; Yu, M.; Yan, X.; Yin, A.An efficient document skew detection method using probability model and q test. *Electronics* **2020**, *9*, 55. [CrossRef]
12. Hussain, S.; Yu, Y.; Ayoub, M.; Khan, A.; Rehman, R.; Wahid, J.A.; Hou, W. IoT and Deep Learning Based Approach for Rapid Screening and Face Mask Detection for Infection Spread Control of COVID-19. *Appl. Sci.* **2021**, *11*, 3495. [CrossRef]
13. Dean, R.B.; Dixon, W.J. Simplified Statistics for Small Numbers of Observations. *Anal. Chem.* **1951**, *23*, 636–638. [CrossRef]
14. Denkena, B.; Bergmann, B.; Stiehl, T.H. Wear curve based online feature assessment for tool condition monitoring. *Procedia CIRP* **2020**, *88*, 312–317. [CrossRef]
15. McBane, G.C. Programs to Compute Distribution Functions and Critical Values for Extreme Value Ratios for Outlier Detection. *J. Stat. Softw.* **2006**, *16*, 1–9. [CrossRef]
16. Gonzalez, T.; Sahni, S.; Franta, W.R. An Efficient Algorithm for the Kolmogorov-Smirnov and Lilliefors Tests. *ACM Trans. Math. Softw.* **1977**, *3*, 60–64. [CrossRef]
17. Lall, A. Data streaming algorithms for the Kolmogorov-Smirnov test. In Proceedings of the International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 95–104.