

Kernel QuantTree (Supplementary Material)

1 Introduction

This document provides additional material omitted from the main article due to space limitations. Section 2 reports the proofs of the theoretical results supporting the Kernel QuantTree (KQT) algorithm, namely, the independence of the test statistic from the stationary distribution (Theorem 1) and the roto-translational invariance of KQT (Theorem 2). Then, Section 3 illustrates additional experimental settings that complete the empirical analysis of the KQT algorithm. In particular, we test the normality of the employed real-world datasets (Section 3.1), investigate the performance on high-dimensional datasets (Section 3.2), compare the performance of the proposed centroid selection strategies (Section 3.3), and report the complete results of the experiments from the main article (Section 3.4).

2 Theoretical Results

In this section, we report the proofs of the theorems introduced in Section 5 of the main article. To make this section self-contained and improve the overall readability, we recall some definitions that were already introduced in the article.

2.1 Controlling the False Alarm Rate

The Generalized QuantTree (GQT) histogram $h = \{(S_k, \hat{\pi}_k)\}$ partitions the input space \mathbb{R}^d such that the probability $\hat{\pi}_k$ of a stationary sample $\mathbf{x} \sim \phi_0$ to fall in bin S_k is close to a target probability π_k provided as an input parameter. During testing, GQT monitors batches W of ν samples by computing a test statistic \mathcal{T}_h whose value only depends on the number of samples of W falling in each bin. Then, the test statistic is compared against a detection threshold $\tau \in \mathbb{R}$, and a change is detected when

$$\mathcal{T}_h(W) > \tau. \tag{1}$$

A peculiarity of GQT is that each bin S_K is defined as a subset of the sublevel set of a *measurable* kernel function $f_k : \mathbb{R}^d \rightarrow \mathbb{R}$. In this section, we prove Theorem 5.1 of the main article, which we recall here:

Theorem 1. *Let $h = \{(S_k, \hat{\pi}_k)\}_{k=1}^K$ be a Generalized QuantTree histogram constructed using measurable functions $f_k : \mathbb{R}^d \rightarrow \mathbb{R} \forall k$. Let \mathcal{T}_h be a statistic defined over batches W such that $\mathcal{T}_h(W)$ only depends on the number of samples y_1, \dots, y_K of W falling in the bins of h . Then, the distribution of \mathcal{T}_h over stationary batches $W \sim \phi_0$ depends only on the batch size ν , the number of training points N and target probabilities $\{\pi_k\}_k$.*

Theorem 1 implies that the distribution of \mathcal{T}_h computed over stationary batches by a GQT is independent of ϕ_0 , d or $\{f_k\}$, thus allowing us to empirically estimate its distribution and compute a threshold τ such that the False Positive Rate (FPR) achieved by GQT is controlled. The threshold computation strategy is presented in Section 5.2 of the main article. Theorem 1 is a generalization of Theorem 1 from [Boracchi et al., 2018] and its proof follows the same structure based on three propositions. Here, we prove the first of these propositions:

Proposition 1. Let $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$ be i.i.d. realizations of a continuous random vector \mathbf{X} defined over $\mathcal{D} \subset \mathbb{R}^d$. Let $f: \mathbb{R}^d \rightarrow \mathbb{R}$ be a measurable function, and let $Z = f(X)$. We denote with $z_{(1)} \leq z_{(2)} \leq \dots \leq z_{(M)}$ the sorted images of $\{\mathbf{x}_j\}$ through f . For any $L \in \{1, 2, \dots, M\}$, we define the sublevel sets

$$Q_{f,L} := \{\mathbf{x} \in \mathcal{D} : f(\mathbf{x}) \leq z_{(L)}\}. \quad (2)$$

Then, the random variable $p = P_{\mathbf{X}}(Q_{f,L})$ is distributed as $\text{Beta}(L, M - L + 1)$.

Proof. We prove the proposition by showing that p is an order statistic of the uniform distribution, which in turn follows a Beta distribution [Lehmann et al., 2005]. Since f is a measurable function for the considered probability space and X is a continuous random variable (r.v.) in \mathbb{R}^d , by the properties of continuous r.v. [Papoulis and Pillai, 2002], we have that $Z = f(X)$ is also a continuous r.v. in \mathbb{R} . Then, we define $U = F_Z(Z)$, where F_Z is the cdf of Z . Since F_Z is monotonically non-decreasing, we can also define the inverse cdf as:

$$F_Z^{-1}(t) = \inf \{z \in \mathbb{R} \mid F_Z(z) \geq t\}. \quad (3)$$

Then, we have that

$$\begin{aligned} F_U(u) &= P_U(U \leq u) = P_Z(F_Z(Z) \leq u) = \\ &= P_Z(Z \leq F_Z^{-1}(u)) = F_Z(F_Z^{-1}(u)) = u, \end{aligned} \quad (4)$$

hence U is a uniform random variable, since its cumulative density function is the identity. Recall that we assumed that \mathbf{X} is defined over \mathcal{D} , i.e., $P_{\mathbf{X}}(\mathbb{R} \setminus \mathcal{D}) = 0$. Then, exploiting (4), we can express p as follows:

$$\begin{aligned} p &= P_X(Q_{f,L}) = P_X(\mathbf{x} \in \mathcal{D} \mid f(\mathbf{x}) \leq z_{(L)}) = \\ &= P_Z(z \in \mathbb{R} \mid z \leq z_{(L)}) = \\ &= P_U(u \in [0, 1] \mid u \leq u_{(L)}) = u_{(L)}, \end{aligned} \quad (5)$$

where we define $u_{(L)} = F_Z(z_{(L)})$. From (5), we have that p is the L -th order statistic of M samplings of the uniform distribution, and its distribution is $\text{Beta}(L, M - L + 1)$ [Balakrishnan and Rao, 1998]. \square

We refer the reader to [Boracchi et al., 2018] for a thorough description of the derivation of the proof of Theorem 1 from Proposition 1.

2.2 Centroid Selection and Invariance to Roto-Translation

Kernel QuantTree (KQT) defines a partition of the input space by iteratively splitting it in bins S_k that match a target probability, as shown in Figure 2 of the main article. The KQT bins are defined as subsets of sublevel sets of the adopted measurable kernel functions f_k . We report here the formal definition of the KQT histogram bins:

$$\begin{cases} S_1 = \{\mathbf{x} \in \mathbb{R}^d \mid f_1(\mathbf{x}) \leq q_1\} \\ S_k = \{\mathbf{x} \in \bigcap_{j < k} \overline{S_j} \mid f_k(\mathbf{x}) \leq q_k\} \text{ for } k < K \\ S_K = \mathbb{R}^d \setminus \bigcup_{j < K} S_j \end{cases}, \quad (6)$$

where $\overline{S_j}$ denotes the complement of S_j in \mathbb{R}^d , and q_k is the split value computed as a quantile of the training samples projected via f_k .

Section 4.2 of the main article illustrates the kernel functions $f_k: \mathbb{R}^d \rightarrow \mathbb{R}$ adopted by KQT, which are defined as distances from a selected *centroid* $\mathbf{c}_k \in \text{TR}$:

$$f_k(\mathbf{x}) = (\mathbf{x} - \mathbf{c}_k)^\top A(\mathbf{x} - \mathbf{c}_k), \quad (7)$$

where $A \in \mathbb{R}^{d \times d}$ is the *kernel matrix*, which determines the employed distance. In our experiments, we construct KQT using the Euclidean, the Mahalanobis, and the Weighted Mahalanobis [Tipping, 1999] distances. The corresponding kernel matrices are $A = \mathbb{I}_d$ for the Euclidean distance, $A = \text{cov}[\text{TR}]^{-1}$ for the Mahalanobis distance and

$$A = \frac{\sum_{m=1}^M \rho_m \cdot i_m(\mathbf{x}, \mathbf{c}) \cdot C_m^{-1}}{\sum_{m=1}^M \rho_m \cdot i_m(\mathbf{x}, \mathbf{c})} \quad (8)$$

for the Weighted Mahalanobis distance, where μ_m , C_m , and ρ_m denote the mean, covariance matrix, and mixing probability of the m -th Gaussian component of a GMM fitted to TR, and the term $i_m(\mathbf{x}, \mathbf{c})$ approximates the integral over the path from \mathbf{x} to \mathbf{c} with respect to the measure induced by the Gaussian Mixture Model (GMM). We refer the reader to [Tipping, 1999] for an explanation of the rationale behind this distance.

2.2.1 Selecting Centroids by Maximizing the Information Gain

In Section 4.3 of the paper, we propose a centroid selection strategy that consists in maximizing the information gain introduced by the split that divides \mathcal{X}_{k-1} in \mathcal{X}_k and $\overline{\mathcal{X}}_k = \mathcal{X}_{k-1} \setminus \mathcal{X}_k$, namely:

$$\mathbf{c}_k = \underset{\mathbf{c} \in \mathcal{X}_{k-1}}{\text{argmax}} I[\mathbf{c}] = \underset{\mathbf{c} \in \mathcal{X}_{k-1}}{\text{argmax}} \left\{ H(\mathcal{X}_{k-1}) - \frac{|\overline{\mathcal{X}}_k| H(\overline{\mathcal{X}}_k) + |\mathcal{X}_k| H(\mathcal{X}_k)}{|\mathcal{X}_{k-1}|} \right\}, \quad (9)$$

where $H(B)$ is the entropy of a set of points $B \subset \mathbb{R}^d$, which we compute by its Gaussian approximation, that is

$$H(B) = (1/2) \log \left((2\pi e)^d \det(\text{cov}[B]) \right), \quad (10)$$

where e is Euler's number. This approximation is only used to ease the computation of $H(B)$ for centroid selection purposes, and does not influence the non-parametric nature of KQT. The expression in (10) can be reformulated:

$$H(B) = \frac{d}{2} (\log(2\pi) + 1) + \tilde{H}(B) \quad (11)$$

where $\tilde{H}(B) = \log \det(\text{cov}[B])$. This gives rise to an optimization problem equivalent to (9), where the centroid is selected by

$$\mathbf{c}_k = \underset{\mathbf{c} \in \mathcal{X}_{k-1}}{\text{argmin}} \left\{ \tilde{H}(\overline{\mathcal{X}}_k) + \beta \tilde{H}(\mathcal{X}_k) \right\}, \quad (12)$$

where β is a constant that can be derived by (9) through algebraic manipulation. Solving this minimization problem is computationally less demanding than the original maximization, thus lowering the computational burden of such centroid selection strategy.

2.2.2 Invariance to roto-translations

In Section 5.3 of the main article, we state that KQT is invariant under roto-translations when the employed kernel function is either the Euclidean, Mahalanobis or Weighted Mahalanobis distance. Here, we prove it together with an intermediate result. In the following, we define a roto-translation $\Phi: \mathbb{R}^d \rightarrow \mathbb{R}^d$ as

$$\Phi(\mathbf{x}) = R(\mathbf{x} - \boldsymbol{\mu}), \quad (13)$$

where $R \in SO(d)$ is the rotation matrix and $\boldsymbol{\mu} \in \mathbb{R}^d$ is the shift vector. Moreover, we denote as $\Phi(B) = \{\Phi(\mathbf{x}) \mid \mathbf{x} \in B\}$ the image of a set $B \subset \mathbb{R}^d$. From basic calculus, it is easy to show that the covariance of a set $B \subset \mathbb{R}^d$ after roto-translation Φ factorizes as

$$\text{cov}[\Phi(B)] = R \text{cov}[B] R^\top. \quad (14)$$

Moreover, in our discussion, we will denote as $D : \mathbb{R}^d \rightarrow \mathbb{R}^d$ the distance employed by KQT when no preprocessing is employed, while we denote as D' the same distance when data are preprocessed by a roto-translation Φ . D and D' coincide when we employ the Euclidean distance, where A is simply the identity matrix. However, the kernel matrices for the Mahalanobis and Weighted Mahalanobis distances depend on the training set TR, thus change when we transform it to $\text{TR}' = \Phi(\text{TR})$. Nevertheless, all the considered distances are invariant under roto-translation, namely it holds that

$$D(\mathbf{x}, \mathbf{y}) = D'(\Phi(\mathbf{x}), \Phi(\mathbf{y})) \quad (15)$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. The identity in (15) can be derived from algebraic manipulation of the definition of the adopted distances and considering (14).

Theorem 5.5 of the main article, which we report here, states that the histograms $h = \{(S_k, \hat{\pi}_k)\}$ and $h' = \{(S'_k, \hat{\pi}'_k)\}$, respectively constructed by KQT with and without preprocessing TR by Φ , are equivalent.

Theorem 2. *Let $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be a roto-translation. Let $h = \{(S_k, \hat{\pi}_k)\}$ and $h' = \{(S'_k, \hat{\pi}'_k)\}$ be the KQT histograms constructed from the training sets $\text{TR} \subset \mathbb{R}^d$ and $\text{TR}' = \Phi(\text{TR})$, where the employed kernel function is either the Euclidean, Mahalanobis or Weighted Mahalanobis distance. Then, we have that $S'_k = \Phi(S_k)$ and $\hat{\pi}'_k = \hat{\pi}_k$ for $k = 1 \dots, K$. In particular, for any batch W and $W' = \Phi(W)$, we have that $\mathcal{T}_h(W) = \mathcal{T}_{h'}(W')$.*

Theorem 2 proves that, for specific choices of kernel functions, the value of the test statistic computed over a batch W of data does not change if we employ a roto-translation-based preprocessing. As such, KQT does not require preprocessing by PCA, which is sometimes necessary for QT to achieve good detection performance. To prove the theorem, we first prove an intermediate result regarding the centroid selection:

Lemma 1 (Information Gain). *Let \mathcal{X}_{k-1} and $\mathcal{X}'_{k-1} = \Phi(\mathcal{X}_{k-1})$ be the set of points used to construct the KQT histogram bins S_k and S'_k , respectively. Then, the centroid selection by maximizing the information gain as in (9) results in centroids \mathbf{c}_k and $\mathbf{c}'_k = \Phi(\mathbf{c}_k)$.*

Proof. As showed in Section 2.2.1, maximizing (9) is equivalent to minimizing (12). Let $\mathbf{c} \in \mathcal{X}_{k-1}$ be an available training sample, then there exists $\mathbf{c}' = \Phi(\mathbf{c}) \in \mathcal{X}'_{k-1}$. If we assume that \mathcal{X}'_k is the set of training samples falling in S'_k when we use \mathbf{c} as a centroid, we have that

$$\begin{aligned} \mathcal{X}'_k &= \{\mathbf{x}' \in \mathcal{X}'_{k-1} \mid D'(\mathbf{x}', \mathbf{c}') \leq q'_k\} = \\ &= \{\Phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_{k-1}, D'(\Phi(\mathbf{x}), \Phi(\mathbf{c})) \leq q'_k\} = \\ &= \{\Phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_{k-1}, D(\mathbf{x}, \mathbf{c}) \leq q_k\} = \\ &= \Phi(\{\mathbf{x} \in \mathcal{X}_{k-1} \mid D(\mathbf{x}, \mathbf{c}) \leq q_k\}) = \Phi(\mathcal{X}_k), \end{aligned} \quad (16)$$

where we used (15) to substitute q'_k with q_k . Analogously, we have that $\overline{\mathcal{X}'_k} = \Phi(\overline{\mathcal{X}_k})$. Then, from (12), we have that

$$\begin{aligned} \tilde{H}(\mathcal{X}'_k) &= \log \det (\text{cov}[\mathcal{X}'_k]) = \\ &= \log \det (R \text{cov}[\mathcal{X}_k] R^\top) = \\ &= \log \det (\text{cov}[\mathcal{X}_k]) + 2 \log \det (R) = \tilde{H}(\mathcal{X}_k) + \gamma, \end{aligned} \quad (17)$$

where γ is a constant which depends only on R . In (17), we used the factorization (14) and the fact

that R is orthogonal. The same relation holds for $\bar{\mathcal{X}}'_k$, and we can finally prove that

$$\begin{aligned}
\mathbf{c}'_k &= \operatorname{argmin}_{\mathbf{c} \in \mathcal{X}'_{k-1}} \left\{ \tilde{H}(\bar{\mathcal{X}}'_k) + \beta \tilde{H}(\mathcal{X}'_k) \right\} = \\
&= \Phi \left(\operatorname{argmin}_{\mathbf{c} \in \mathcal{X}_{k-1}} \left\{ \tilde{H}(\bar{\mathcal{X}}_k) + \beta \tilde{H}(\mathcal{X}_k) + \gamma(1 + \beta) \right\} \right) = \\
&= \Phi \left(\operatorname{argmin}_{\mathbf{c} \in \mathcal{X}_{k-1}} \left\{ \tilde{H}(\bar{\mathcal{X}}_k) + \beta \tilde{H}(\mathcal{X}_k) \right\} \right) = \Phi(\mathbf{c}_k).
\end{aligned} \tag{18}$$

□

Lemma 2 (Gini Index). *Let \mathcal{X}_{k-1} and $\mathcal{X}'_{k-1} = \Phi(\mathcal{X}_{k-1})$ be the set of points used to construct the KQT histogram bins S_k and S'_k , respectively. Then, the centroid selection by minimizing the Gini index in results in centroids \mathbf{c}_k and $\mathbf{c}'_k = \Phi(\mathbf{c}_k)$.*

Proof. It can be shown by simple algebraic manipulation of the definition of Gini index. □

Lemma 1 and Lemma 2 ensure that the construction of the histograms h and h' will maintain the correspondance through Φ of all their elements, including the selected centroids. We can now prove Theorem 2.

Proof of Theorem 2. Here, we show by induction that every bin S'_k of h' is the result of the roto-translation of the corresponding bin S_k of h . First, we have that $\mathcal{X}'_0 = \operatorname{TR}' = \Phi(\operatorname{TR}) = \Phi(\mathcal{X}_0)$ by definition. Then, for $k = 1$, Lemma 1 and 2 state that $\mathbf{c}'_1 = \Phi(\mathbf{c}_1)$. Moreover,

$$\begin{aligned}
S'_1 &= \{ \mathbf{x}' \in \mathbb{R}^d \mid D'(\mathbf{x}', \mathbf{c}'_1) \leq q'_1 \} = \\
&= \{ \Phi(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^d, D'(\Phi(\mathbf{x}), \mathbf{c}'_1) \leq q'_1 \} = \\
&= \{ \Phi(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^d, D(\mathbf{x}, \mathbf{c}_1) \leq q_1 \} = \Phi(S_1).
\end{aligned} \tag{19}$$

In the same manner, we prove that

$$\begin{aligned}
X'_1 &= \{ \mathbf{x}' \in \mathcal{X}'_0 \mid D'(\mathbf{x}', \mathbf{c}'_1) > q'_1 \} = \\
&= \{ \Phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_0, D(\mathbf{x}, \mathbf{c}_1) > q_1 \} = \Phi(X_1),
\end{aligned} \tag{20}$$

and $\bar{\mathcal{X}}'_1 = \Phi(\bar{\mathcal{X}}_1)$.

Now, suppose that $\forall j < k$ we have that $\mathbf{c}'_j = \Phi(\mathbf{c}_j)$, $S'_j = \Phi(S_j)$ and $\mathcal{X}'_j = \Phi(\mathcal{X}_j)$. Then, we have that $\mathbf{x} \in \bigcap_{j < k} \bar{S}_j \iff \mathbf{x}' = \Phi(\mathbf{x}) \in \bigcap_{j < k} \bar{S}'_j$, and, with the same derivation as in the case $k = 1$,

$$\begin{aligned}
S'_k &= \left\{ \mathbf{x}' \in \bigcap_{j < k} \bar{S}'_j \mid D'(\mathbf{x}', \mathbf{c}'_k) \leq q_k \right\} = \\
&= \Phi \left(\left\{ \mathbf{x} \in \bigcap_{j < k} \bar{S}_j \mid D(\mathbf{x}, \mathbf{c}_k) \leq q_k \right\} \right) = \Phi(S_k).
\end{aligned} \tag{21}$$

and also $\mathcal{X}'_k = \Phi(\mathcal{X}_k)$. In conclusion, we proved that $S'_k = \Phi(S_k)$ for $\forall k = 1, \dots, K$. In particular, we conclude that

$$\begin{aligned}
\mathbf{x} \in S_k &\iff D(\mathbf{x}, \mathbf{c}_j) > q_j \forall j < k \wedge D(\mathbf{x}, \mathbf{c}_k) \leq q_k \iff \\
&\iff D'(\Phi(\mathbf{x}), \mathbf{c}'_j) > q_j \forall j < k \wedge D'(\Phi(\mathbf{x}), \mathbf{c}'_k) \leq q_k \iff \\
&\iff \Phi(\mathbf{x}) \in S'_k,
\end{aligned} \tag{22}$$

and, consequently, the number of samples from any batch $W \subset \mathbb{R}^d$ falling in the S_k is the same as the number of samples of $W' = \Phi(W)$ falling in S'_k . Then, we have that $\hat{\pi}_k = \hat{\pi}'_k$ and $\mathcal{T}_h(W) = \mathcal{T}_{h'}(W')$. □

3 More Experiments and Discussion

This section extends the experimental evaluation of KQT from Section 6 of the main article to corroborate the findings discussed there. First, we investigate the real-world datasets employed in our experiments, proving that these do not follow a Gaussian distribution. Then, we perform additional experiments on high-dimensional data to investigate the control of the FPR in this challenging scenario. Finally, we extend the results from the main article by comparing the proposed centroid selection strategies and reporting the complete results for both the low- and high-ratio settings.

3.1 Remarks about the real-world datasets

In Section 6.1 of the main article, we introduce the real-world datasets that are used in our experiments. The INSECTS dataset [Souza et al., 2020] is a benchmark for concept-drift detection algorithms and comprises data describing the wing-beat frequency of six species of insects at different temperatures. The other datasets are from the UCI Machine Learning Repository [Dua and Graff, 2017] and from [Dal Pozzolo et al., 2017], and comprise data following a unique distribution, thus require the introduction of artificial distribution changes for our experiments. We standardize these datasets and add a negligible amount of noise $\eta \sim N(0, \sigma)$ to each component to prevent the many repeated values from harming the histogram construction. Table 1 lists all the datasets and reports their dimension d and the level σ of noise applied to their components.

Since data in the synthetic settings are drawn from Gaussian distributions, one could argue that KQT provided with the Mahalanobis or Weighted Mahalanobis kernels have an advantage over the alternatives. However, this is not true for the real-world datasets considered in our experiments, which are far from Gaussian. This claim is empirically supported by the low detection performance of SPLL, which is itself based on a GMM. To confirm this intuition, we also run the Shapiro-Wilk normality test [Shapiro and Wilk, 1965], an Hypothesis Test used to determine whether a population $\{x_i\}_{i=1}^n \subset \mathbb{R}$ is drawn from a univariate Gaussian distribution. If the p -values associated to the HT is lower than 0.05, than we can conclude that the population is not normally distributed. Since the marginals of a multivariate Gaussian distribution are univariate Gaussian distributions, we show that the real-world datasets introduced in Section 6.1 of the main article are not drawn from multivariate Gaussians by showing that their covariates are not. For this purpose, we extract a subset of $n = 4096$ samples from the real-world datasets and perform the Shapiro-Wilk test on each of their covariates. Table 1 reports the p -values yielded by the test, averaged over the covariates and over 250 iteration of the test performed over different subsets. The p -values obtained in these tests are in the range of 10^{-20} , thus confirming that the real-world datasets employed in our experiments are not drawn from multivariate Gaussian distributions.

Table 1: List of the real-world datasets employed in the experiments. For each dataset, we report the dimension d , the noise level σ and the average p -value of the Shapiro-Wilk test computed on the marginals.

Dataset	Name	d	σ	p -value	Reference
El Nino Southern Oscillation	nino	5	10^{-3}	3.3×10^{-3}	[Dua and Graff, 2017]
Physicochemical Properties of PTS	protein	9	—	7.5×10^{-8}	[Dua and Graff, 2017]
ForestCovertype I	spruce	10	10^{-1}	2.5×10^{-9}	[Dua and Graff, 2017]
ForestCovertype II	lodgepole	10	10^{-1}	1.8×10^{-8}	[Dua and Graff, 2017]
Credit Card Fraud Detection	credit	28	10^{-3}	9.9×10^{-8}	[Dal Pozzolo et al., 2017]
Insects' Flying Behavior	INSECTS	33	—	$< 10^{-16}$	[Souza et al., 2020]
Sensorless Drive Diagnosis	sensorless	48	10^{-3}	4.9×10^{-8}	[Dua and Graff, 2017]
MiniBooNE Particle Identification	particle	50	10^{-3}	8.5×10^{-3}	[Dua and Graff, 2017]
UNSW Swarm Behavior	swarm	2400	—	1.8×10^{-9}	[Dua and Graff, 2017]

3.2 Curse of dimensionality

In this section, we investigate the ability of KQT to control the FPR as the data dimension d increases. Our experiments (Section 6.4 of the main article) have shown that the Kernel QuantTree with the Weighted Mahalanobis distance deviates from the desired FPR when the data dimension grows. As discussed in the article, this deviation is due to the challenge of fitting a Gaussian Mixture Model (GMM) to high-dimensional data. To analyze the impact of the dimensionality on Kernel QuantTree, we perform experiments in three synthetic settings with $d \in \{4, 8, 16, 32, 64, 128\}$. In these settings, denoted as *unimodal*, *bimodal*, and *trimodal*, the stationary distribution ϕ_0 is defined as a GMM with 1, 2, and 3 Gaussian components, respectively. Then, we use the CCM framework [Alippi et al., 2017] to generate a post-change distribution by applying a roto-translation to each Gaussian component of ϕ_0 such that the Kullback-Leibler distance between these and the three resulting post-change components is fixed to 1. We perform each experiment twice, one with $N = 4096$ training samples and the other with $N = 16384$, to show that when a large training set is available the limitation of the KQT with Weighted Mahalanobis is avoidable.

Table 2 reports the FPR achieved by KQT adopting different distances in the unimodal, bimodal and trimodal settings for all the considered dimensions d and training set sizes N . In all experiments we construct a KQT histogram with $K = 16$ bins, we set the detection thresholds to yield an FPR $\alpha = 5\%$, and we test KQT on 5000 stationary batches with $\nu = 128$ samples. In the experiment with $N = 4096$ (left columns), we notice that when d increases, the FPR achieved by KQT when using the Mahalanobis and Weighted Mahalanobis distances deviates further from the target value. In contrast, when we train KQT on $N = 16384$ samples (right columns), the deviation from the target FPR is significantly reduced. To further corroborate our hypothesis that the issue is in the GMM fitting, we compute the average condition number of the covariance matrices of the GMM components yielded when using KQT with the Weighted Mahalanobis distance. These results show that, in the high-dimensional datasets, using a larger training set yields covariance matrices with smaller condition numbers.

3.3 Comparing the centroid selection strategies

In the main article, we propose two strategies for the centroid selection, namely, maximizing the information gain introduced by splitting \mathcal{X}_{k-1} in \mathcal{X}_k and $\bar{\mathcal{X}}_k$ and minimizing the Gini index of the distances between the centroid and the training samples in \mathcal{X}_{k-1} . In Table 1 of the article, we report the average FPR and AUC achieved by KQT using the maximization of the information gain as a centroid selection strategy. Here, Table 3 reports the results achieved by KQT with the Euclidean, Mahalanobis, and Weighted Mahalanobis distance for both strategies and proves that their performance is comparable in every experimental setting.

3.4 Complete experimental results

In this section, we report the complete results of the high- and low-ratio experiments presented in Section 6 of the main article. For each result, we include the corresponding confidence interval.

Table 5 reports the FPR and AUC achieved by the methods presented in Section 6.3 of the main article in the high-ratio setting, namely when $\nu = 128$ and $K = 16$. As already discussed in the paper, QuantTree, Kernel QuantTree and EIKM achieve an empirical FPR close to the target $\alpha = 5\%$ in most experiments. In contrast, SPLL and PCA-SPLL mostly exceed the target and Density Tree largely overshoots it. However, the KQT with the Weighted Mahalanobis distance does not control the FPR accurately when d increases, and we speculate that this is due to the GMM underlying the definition of distance. This known limitation is discussed in Section 6.4 of the main article and investigated in Section 3.2 of this document.

As for the AUC, KQT with the Weighted Mahalanobis distances outperforms the alternatives in most settings. At the bottom of Table 5, we report the ranking of each method computed from the

Table 2: Comparison between the FPR achieved by KQT using the Euclidean, Mahalanobis, and Weighted Mahalanobis distances in the synthetic settings for various dimensions d and training set sizes N . In parenthesis, the average condition numbers of the covariance matrices of the GMM used by KQT with the Weighted Mahalanobis distance. The underlined values indicate an FPR above the target of 5%.

	KQT(Euclidean)		KQT(Mahalanobis)		KQT(Weighted Mahalanobis)		
	N=4096	N=16384	N=4096	N=16384	N=4096	N=16384	
$d = 4$	4.88%	-	4.77%	4.84%	4.79% (20.1)	4.85% (16.3)	unimodal
$d = 8$	4.83%	-	4.81%	4.86%	4.71% (32.1)	4.83% (37.7)	
$d = 16$	4.81%	-	4.81%	4.89%	4.88% (99.5)	4.79% (67.7)	
$d = 32$	4.84%	-	4.95%	4.88%	4.99% (150.6)	4.81% (123.6)	
$d = 64$	4.84%	-	<u>5.80%</u>	4.95%	<u>5.81%</u> (315.2)	4.87% (223.8)	
$d = 128$	4.91%	-	<u>16.52%</u>	<u>5.31%</u>	<u>77.74%</u> (344.0)	<u>5.45%</u> (307.0)	
$d = 4$	4.83%	-	4.76%	4.88%	4.77% (12.9)	4.89% (13.7)	bimodal
$d = 8$	4.88%	-	4.86%	4.87%	4.79% (40.0)	4.84% (36.3)	
$d = 16$	4.83%	-	4.88%	4.82%	4.88% (68.3)	4.87% (90.3)	
$d = 32$	4.86%	-	4.95%	4.86%	<u>5.36%</u> (177.2)	4.83% (120.4)	
$d = 64$	4.89%	-	<u>5.66%</u>	4.86%	<u>5.70%</u> (253.9)	<u>5.03%</u> (220.3)	
$d = 128$	4.84%	-	<u>15.44%</u>	<u>5.32%</u>	<u>76.60%</u> (276.9)	<u>5.46%</u> (244.7)	
$d = 4$	4.72%	-	4.86%	4.85%	4.82% (24.9)	4.84% (16.1)	trimodal
$d = 8$	4.84%	-	4.79%	4.82%	4.83% (31.7)	4.80% (38.2)	
$d = 16$	4.85%	-	4.85%	4.80%	4.86% (66.1)	4.83% (59.9)	
$d = 32$	4.81%	-	4.91%	4.84%	<u>5.13%</u> (108.7)	4.86% (120.7)	
$d = 64$	4.93%	-	<u>5.67%</u>	4.87%	<u>5.53%</u> (209.4)	<u>5.01%</u> (176.9)	
$d = 128$	4.81%	-	<u>15.86%</u>	<u>5.37%</u>	<u>77.49%</u> (258.1)	<u>5.47%</u> (214.2)	

AUC, together with the p -value of the Nemenyi post-hoc statistic, which proves that the advantage of the best-performing method is statistically significant. In the main article, we also discuss the performance of QuantTree, which shows how the preprocessing by PCA decreases the detection performance in some cases. Remarkably, the KQT monitoring is invariant under roto-translations (see Section 5.3 of the main article) and surpasses QuantTree independently of the application of the PCA preprocessing.

Table 4 reports the FPR and AUC achieved by the considered methods in the low-ratio setting, namely when $\nu = 64$ and $K = 32$. The results of this experiment are overall in line with the high-ratio setting. However, as we speculate in Section 6.3 of the main article, histogram can better model a data distribution when the expected number of points per bin ν/K is large. This low-ratio setting confirms our speculation, as the considered methods achieve an AUC lower than in the high-ratio experiment on most datasets. However, KQT with the Weighted Mahalanobis distance still achieves the best AUC with a statistically significant advantage over the alternatives, as demonstrated by the Nemenyi post-hoc test. Moreover, the Pearson test statistic is discrete and in the low-ratio setting assumes fewer distinct values. Thus, it is more challenging to set detection thresholds and the results show that the empirical FPR of QuantTree and Kernel QuantTree is slightly lower than in the high-ratio experiment.

Table 3: Comparison between the detection performance achieved by KQT with the Euclidean, Mahalanobis and Weighted Mahalanobis distances, when selecting the centroids by maximization of the Information Gain (left) and minimization of the Gini Index (right), in the high-ratio setting ($K = 16$, $\nu = 128$ points). The table reports the achieved FPR (top) and AUC (bottom). In parenthesis the standard deviation.

	Information Gain			Gini Index		
	Euclidean	Mahalanobis	Weighted Maha.	Euclidean	Mahalanobis	Weighted Maha.
unimodal	4.86% (0.47%)	4.82% (0.45%)	4.83% (0.48%)	4.83% (0.47%)	4.81% (0.48%)	4.83% (0.49%)
bimodal	4.80% (0.46%)	4.81% (0.44%)	4.80% (0.45%)	4.81% (0.47%)	4.84% (0.46%)	4.83% (0.46%)
nino	5.00% (0.53%)	5.02% (0.53%)	5.01% (0.54%)	5.02% (0.55%)	5.06% (0.54%)	5.02% (0.54%)
protein	4.97% (0.52%)	4.98% (0.54%)	5.03% (0.55%)	4.99% (0.54%)	5.03% (0.56%)	5.06% (0.53%)
spruce	4.82% (0.47%)	4.84% (0.49%)	4.90% (0.47%)	4.84% (0.49%)	4.85% (0.48%)	4.88% (0.49%)
lodgepole	4.85% (0.49%)	4.80% (0.47%)	4.90% (0.50%)	4.84% (0.50%)	4.82% (0.49%)	4.93% (0.51%)
credit	4.89% (0.48%)	4.85% (0.46%)	5.06% (0.56%)	4.89% (0.50%)	4.90% (0.52%)	5.10% (0.61%)
insects (1 → 2)	4.91% (0.50%)	4.93% (0.52%)	5.19% (0.64%)	4.90% (0.49%)	4.92% (0.54%)	5.19% (0.61%)
insects (2 → 3)	4.92% (0.52%)	4.96% (0.52%)	5.25% (0.62%)	4.88% (0.50%)	4.93% (0.52%)	5.24% (0.63%)
insects (3 → 4)	4.90% (0.52%)	4.88% (0.53%)	5.22% (0.64%)	4.91% (0.55%)	4.92% (0.53%)	5.24% (0.64%)
insects (4 → 5)	4.91% (0.51%)	4.92% (0.54%)	5.25% (0.65%)	4.92% (0.52%)	4.95% (0.49%)	5.28% (0.66%)
insects (5 → 6)	4.90% (0.56%)	4.92% (0.53%)	5.26% (0.72%)	4.90% (0.55%)	4.91% (0.57%)	5.29% (0.71%)
sensorless	4.82% (0.49%)	5.01% (0.56%)	7.42% (1.61%)	4.87% (0.48%)	4.98% (0.58%)	7.54% (1.56%)
particle	4.81% (0.46%)	4.94% (0.52%)	5.80% (1.02%)	4.84% (0.48%)	4.93% (0.52%)	5.86% (1.00%)
unimodal	0.946 (0.105)	0.993 (0.016)	0.994 (0.013)	0.946 (0.103)	0.994 (0.015)	0.994 (0.014)
bimodal	0.904 (0.118)	0.954 (0.060)	0.968 (0.042)	0.903 (0.119)	0.955 (0.056)	0.970 (0.039)
nino	0.607 (0.072)	0.904 (0.138)	0.922 (0.122)	0.609 (0.071)	0.903 (0.139)	0.922 (0.122)
protein	0.617 (0.074)	0.993 (0.035)	0.995 (0.027)	0.615 (0.074)	0.993 (0.030)	0.994 (0.030)
spruce	0.601 (0.066)	1.000 (0.000)	1.000 (0.000)	0.600 (0.068)	1.000 (0.000)	1.000 (0.000)
lodgepole	0.654 (0.099)	1.000 (0.000)	1.000 (0.000)	0.653 (0.099)	1.000 (0.000)	1.000 (0.000)
credit	0.602 (0.053)	0.780 (0.146)	1.000 (0.000)	0.605 (0.055)	0.787 (0.141)	1.000 (0.000)
insects (1 → 2)	0.962 (0.035)	0.972 (0.039)	0.993 (0.019)	0.961 (0.038)	0.970 (0.037)	0.994 (0.015)
insects (2 → 3)	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)
insects (3 → 4)	0.904 (0.054)	0.942 (0.049)	0.990 (0.024)	0.903 (0.058)	0.946 (0.044)	0.989 (0.025)
insects (4 → 5)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
insects (5 → 6)	0.985 (0.016)	0.992 (0.008)	1.000 (0.000)	0.986 (0.014)	0.991 (0.009)	1.000 (0.001)
sensorless	0.542 (0.027)	1.000 (0.000)	1.000 (0.000)	0.543 (0.026)	1.000 (0.000)	1.000 (0.000)
particle	0.555 (0.030)	0.976 (0.051)	0.985 (0.039)	0.556 (0.032)	0.977 (0.051)	0.985 (0.042)

Table 4: FPR (top) and AUC (bottom) achieved by the considered methods in the high-ratio setting, namely when histogram-based methods define $K = 32$ bins and the monitored batches contain $\nu = 64$ points. In parenthesis, the width of the 95%-confidence interval of the results.

	QT	QT (PCA)	KQT (Euclidean)	KQT (Mahalanobis)	KQT (Weighted Maha.)	EIRM	SPLL(C=3)	SPLL(C=3) (PCA)	DT	DT (PCA)
unimodal	4.29% (0.34%)	4.27% (0.32%)	4.30% (0.32%)	4.30% (0.34%)	4.28% (0.33%)	4.91% (0.44%)	6.10% (0.66%)	7.24% (1.03%)	7.08% (1.01%)	7.06% (1.00%)
bimodal	4.31% (0.32%)	4.27% (0.32%)	4.29% (0.33%)	4.32% (0.33%)	4.29% (0.33%)	4.91% (0.44%)	6.29% (0.79%)	7.57% (1.26%)	6.99% (1.00%)	6.92% (0.97%)
nino	5.01% (0.38%)	5.18% (0.38%)	5.23% (0.37%)	5.18% (0.38%)	5.21% (0.38%)	4.91% (0.48%)	7.28% (1.16%)	9.60% (1.75%)	6.77% (0.98%)	6.86% (1.09%)
protein	4.98% (0.38%)	5.13% (0.39%)	5.17% (0.39%)	5.19% (0.40%)	5.21% (0.43%)	4.92% (0.47%)	15.38% (3.01%)	10.74% (2.28%)	6.81% (0.98%)	6.73% (1.08%)
spruce	4.26% (0.33%)	4.29% (0.32%)	4.30% (0.32%)	4.29% (0.33%)	4.29% (0.34%)	4.85% (0.45%)	13.42% (3.15%)	13.64% (3.18%)	6.76% (0.99%)	6.89% (1.08%)
lodgpole	4.27% (0.32%)	4.27% (0.34%)	4.29% (0.35%)	4.30% (0.36%)	4.31% (0.34%)	4.93% (0.52%)	12.57% (3.45%)	12.73% (3.46%)	6.89% (1.00%)	6.74% (1.06%)
credit	4.34% (0.39%)	4.46% (0.46%)	4.39% (0.43%)	4.38% (0.42%)	4.47% (0.44%)	4.91% (0.61%)	11.88% (2.16%)	23.23% (3.66%)	6.74% (1.04%)	6.75% (1.02%)
insects (1 → 2)	4.69% (0.47%)	4.84% (0.57%)	4.84% (0.58%)	4.84% (0.56%)	4.97% (0.59%)	4.91% (0.45%)	6.73% (1.63%)	8.05% (1.95%)	6.86% (1.04%)	6.77% (1.10%)
insects (2 → 3)	4.73% (0.50%)	4.87% (0.59%)	4.85% (0.56%)	4.86% (0.60%)	5.04% (0.61%)	4.90% (0.45%)	6.67% (1.51%)	8.02% (1.77%)	6.83% (1.03%)	6.77% (1.04%)
insects (3 → 4)	4.72% (0.49%)	4.84% (0.56%)	4.85% (0.58%)	4.84% (0.58%)	5.02% (0.60%)	4.93% (0.48%)	7.29% (1.75%)	8.71% (2.03%)	6.80% (1.07%)	6.75% (1.03%)
insects (4 → 5)	4.69% (0.50%)	4.84% (0.58%)	4.82% (0.57%)	4.83% (0.60%)	4.99% (0.64%)	4.90% (0.44%)	6.56% (1.55%)	7.89% (1.87%)	6.83% (1.03%)	6.71% (1.03%)
insects (5 → 6)	4.77% (0.53%)	4.90% (0.56%)	4.90% (0.56%)	4.89% (0.58%)	5.07% (0.60%)	4.86% (0.43%)	7.18% (1.72%)	8.09% (1.89%)	6.77% (1.04%)	6.76% (1.00%)
sensorless	4.29% (0.35%)	4.43% (0.39%)	4.30% (0.34%)	4.35% (0.36%)	5.32% (0.65%)	4.94% (0.49%)	4.93% (1.19%)	4.29% (0.71%)	6.53% (1.06%)	6.67% (1.07%)
particle	4.28% (0.32%)	4.32% (0.36%)	4.30% (0.35%)	4.33% (0.35%)	4.71% (0.46%)	4.86% (0.50%)	6.92% (1.52%)	8.80% (1.96%)	6.70% (1.12%)	6.78% (1.12%)
unimodal	0.883 (0.101)	0.936 (0.059)	0.881 (0.115)	0.957 (0.031)	<u>0.957</u> (0.031)	0.779 (0.157)	0.975 (0.016)	0.972 (0.047)	0.676 (0.148)	0.712 (0.174)
bimodal	0.796 (0.109)	0.825 (0.102)	0.821 (0.112)	0.868 (0.075)	0.885 (0.062)	0.709 (0.130)	0.859 (0.132)	0.845 (0.154)	0.636 (0.106)	0.652 (0.122)
nino	0.736 (0.143)	0.804 (0.170)	0.555 (0.042)	0.809 (0.173)	0.830 (0.163)	0.511 (0.012)	0.739 (0.176)	0.771 (0.191)	0.630 (0.111)	0.545 (0.048)
protein	0.848 (0.104)	0.980 (0.055)	0.582 (0.048)	0.985 (0.050)	0.991 (0.035)	0.808 (0.009)	0.906 (0.118)	0.945 (0.098)	0.638 (0.117)	0.599 (0.081)
spruce	1.000 (0.003)	1.000 (0.000)	0.590 (0.060)	1.000 (0.000)	1.000 (0.000)	0.504 (0.005)	1.000 (0.002)	1.000 (0.002)	1.000 (0.001)	1.000 (0.001)
lodgpole	1.000 (0.001)	1.000 (0.001)	0.639 (0.085)	1.000 (0.000)	1.000 (0.000)	0.506 (0.006)	1.000 (0.002)	1.000 (0.002)	1.000 (0.000)	1.000 (0.000)
credit	0.611 (0.043)	0.813 (0.144)	0.550 (0.021)	0.685 (0.137)	1.000 (0.000)	0.504 (0.005)	0.565 (0.060)	0.624 (0.108)	0.603 (0.051)	0.739 (0.116)
insects (1 → 2)	0.976 (0.022)	0.887 (0.054)	0.902 (0.041)	0.967 (0.025)	0.959 (0.037)	0.698 (0.057)	0.733 (0.033)	0.786 (0.031)	1.000 (0.000)	0.999 (0.002)
insects (2 → 3)	0.968 (0.031)	0.981 (0.018)	0.994 (0.005)	0.999 (0.001)	1.000 (0.001)	0.895 (0.052)	1.000 (0.000)	0.999 (0.000)	0.987 (0.008)	0.996 (0.009)
insects (3 → 4)	0.915 (0.045)	0.804 (0.068)	0.821 (0.046)	0.909 (0.036)	0.940 (0.051)	0.688 (0.066)	0.691 (0.021)	0.687 (0.023)	0.995 (0.003)	0.987 (0.007)
insects (4 → 5)	0.989 (0.015)	0.991 (0.015)	0.998 (0.003)	1.000 (0.001)	1.000 (0.000)	0.878 (0.071)	1.000 (0.000)	1.000 (0.000)	0.999 (0.001)	0.994 (0.010)
insects (5 → 6)	0.974 (0.017)	0.890 (0.044)	0.922 (0.023)	0.961 (0.019)	0.982 (0.011)	0.860 (0.051)	0.932 (0.007)	0.933 (0.008)	0.997 (0.001)	0.996 (0.002)
sensorless	0.832 (0.112)	0.999 (0.008)	0.523 (0.013)	1.000 (0.000)	1.000 (0.000)	0.501 (0.003)	1.000 (0.000)	1.000 (0.000)	0.715 (0.139)	0.582 (0.083)
particle	0.860 (0.129)	0.876 (0.107)	0.530 (0.015)	0.922 (0.101)	0.941 (0.089)	0.503 (0.004)	0.786 (0.143)	0.861 (0.127)	0.706 (0.143)	0.526 (0.035)
Average Ranking	5.32	4.96	7.28	3.78	2.96	9.54	5.26	4.97	5.33	5.60
Nemenyi p -value	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	-	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$

Table 5: FPR (top) and AUC (bottom) achieved by the considered methods in the high-ratio setting, namely when histogram-based methods define $K = 16$ bins and the monitored batches contain $\nu = 128$ points. In parenthesis, the width of the 95%-confidence interval of the results.

	QT	QT (PCA)	KQT (Euclidean)	KQT (Mahalanobis)	KQT (Weighted Maha.)	EIRM	SPLL(C=3)	SPLL(C=3) (PCA)	DT	DT (PCA)
unimodal	4.83% (0.48%)	4.81% (0.46%)	4.86% (0.47%)	4.82% (0.45%)	4.83% (0.48%)	4.82% (0.53%)	5.46% (0.75%)	5.92% (1.04%)	7.84% (1.16%)	7.75% (1.17%)
bimodal	4.80% (0.45%)	4.81% (0.46%)	4.80% (0.46%)	4.81% (0.44%)	4.80% (0.45%)	4.82% (0.51%)	5.53% (0.75%)	6.02% (1.06%)	7.65% (1.20%)	7.62% (1.09%)
nino	5.04% (0.49%)	4.99% (0.50%)	5.00% (0.53%)	5.02% (0.53%)	5.01% (0.54%)	4.83% (0.55%)	6.14% (1.21%)	7.69% (2.05%)	7.55% (1.20%)	7.57% (1.16%)
protein	4.97% (0.50%)	4.98% (0.56%)	4.97% (0.52%)	4.98% (0.54%)	5.03% (0.55%)	4.88% (0.61%)	13.15% (3.54%)	8.42% (2.33%)	7.65% (1.25%)	7.64% (1.25%)
spruce	4.81% (0.50%)	4.83% (0.48%)	4.82% (0.47%)	4.84% (0.49%)	4.90% (0.47%)	4.86% (0.59%)	11.43% (3.93%)	11.56% (3.97%)	7.56% (1.21%)	7.57% (1.16%)
lodgpole	4.83% (0.47%)	4.82% (0.50%)	4.85% (0.49%)	4.80% (0.47%)	4.90% (0.50%)	4.92% (0.57%)	10.78% (4.64%)	10.89% (4.68%)	7.60% (1.14%)	7.58% (1.12%)
credit	4.83% (0.47%)	4.96% (0.54%)	4.89% (0.48%)	4.85% (0.46%)	5.06% (0.56%)	4.96% (0.68%)	8.67% (2.26%)	16.06% (3.63%)	7.63% (1.15%)	7.59% (1.23%)
insects (1 → 2)	4.92% (0.50%)	4.93% (0.51%)	4.91% (0.50%)	4.93% (0.52%)	5.19% (0.64%)	4.93% (0.63%)	5.90% (2.04%)	6.48% (2.15%)	7.57% (1.16%)	7.60% (1.20%)
insects (2 → 3)	4.93% (0.53%)	4.91% (0.54%)	4.92% (0.52%)	4.96% (0.52%)	5.25% (0.62%)	4.96% (0.65%)	5.54% (1.85%)	6.16% (1.96%)	7.60% (1.19%)	7.59% (1.23%)
insects (3 → 4)	4.92% (0.48%)	4.89% (0.52%)	4.90% (0.52%)	4.88% (0.53%)	5.22% (0.64%)	4.89% (0.58%)	6.09% (1.99%)	6.69% (2.11%)	7.59% (1.19%)	7.54% (1.17%)
insects (4 → 5)	4.92% (0.50%)	4.95% (0.52%)	4.91% (0.51%)	4.92% (0.54%)	5.25% (0.65%)	4.91% (0.61%)	5.48% (1.76%)	6.01% (1.84%)	7.63% (1.24%)	7.56% (1.17%)
insects (5 → 6)	4.91% (0.54%)	4.90% (0.54%)	4.90% (0.56%)	4.92% (0.53%)	5.26% (0.72%)	4.90% (0.64%)	5.86% (2.05%)	6.19% (2.11%)	7.61% (1.22%)	7.63% (1.24%)
sensorless	4.84% (0.50%)	5.01% (0.55%)	4.82% (0.49%)	5.01% (0.56%)	7.42% (1.61%)	4.93% (0.61%)	4.33% (1.03%)	4.83% (0.76%)	7.55% (1.19%)	7.58% (1.22%)
particle	4.85% (0.50%)	4.87% (0.51%)	4.81% (0.46%)	4.94% (0.52%)	5.80% (1.02%)	4.84% (0.61%)	5.93% (2.01%)	6.07% (2.05%)	7.52% (1.10%)	7.60% (1.19%)
unimodal	0.957 (0.079)	0.976 (0.057)	0.946 (0.105)	0.993 (0.016)	0.994 (0.013)	0.874 (0.154)	0.996 (0.006)	0.989 (0.040)	0.786 (0.167)	0.806 (0.190)
bimodal	0.900 (0.110)	0.930 (0.090)	0.904 (0.118)	0.954 (0.060)	0.968 (0.042)	0.821 (0.158)	0.915 (0.126)	0.895 (0.164)	0.751 (0.155)	0.767 (0.160)
nino	0.845 (0.143)	0.905 (0.135)	0.607 (0.072)	0.904 (0.138)	0.922 (0.122)	0.528 (0.029)	0.816 (0.172)	0.841 (0.183)	0.726 (0.152)	0.582 (0.081)
protein	0.899 (0.104)	0.985 (0.051)	0.617 (0.074)	0.993 (0.035)	0.995 (0.027)	0.514 (0.015)	0.918 (0.118)	0.954 (0.093)	0.704 (0.148)	0.595 (0.085)
spruce	0.999 (0.014)	1.000 (0.000)	0.601 (0.066)	1.000 (0.000)	1.000 (0.000)	0.507 (0.007)	1.000 (0.000)	1.000 (0.000)	1.000 (0.002)	1.000 (0.002)
lodgpole	1.000 (0.000)	1.000 (0.000)	0.654 (0.099)	1.000 (0.000)	1.000 (0.000)	0.511 (0.016)	1.000 (0.002)	1.000 (0.002)	1.000 (0.000)	1.000 (0.000)
credit	0.698 (0.079)	0.867 (0.127)	0.602 (0.053)	0.780 (0.146)	1.000 (0.000)	0.508 (0.011)	0.597 (0.085)	0.660 (0.132)	0.695 (0.091)	0.820 (0.131)
insects (1 → 2)	0.998 (0.005)	0.962 (0.048)	0.962 (0.035)	0.972 (0.039)	0.993 (0.019)	0.836 (0.071)	0.810 (0.035)	0.866 (0.029)	1.000 (0.000)	1.000 (0.000)
insects (2 → 3)	0.993 (0.017)	0.995 (0.012)	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)	0.962 (0.014)	1.000 (0.000)	1.000 (0.000)	0.999 (0.002)	1.000 (0.001)
insects (3 → 4)	0.983 (0.029)	0.897 (0.078)	0.904 (0.054)	0.942 (0.049)	0.990 (0.024)	0.835 (0.084)	0.753 (0.025)	0.745 (0.028)	1.000 (0.000)	1.000 (0.001)
insects (4 → 5)	0.998 (0.008)	0.997 (0.008)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	0.950 (0.004)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	0.999 (0.004)
insects (5 → 6)	0.999 (0.003)	0.971 (0.033)	0.985 (0.016)	0.992 (0.008)	1.000 (0.000)	0.963 (0.017)	0.979 (0.004)	0.979 (0.005)	1.000 (0.000)	1.000 (0.000)
sensorless	0.862 (0.120)	1.000 (0.004)	0.542 (0.027)	1.000 (0.000)	1.000 (0.000)	0.502 (0.003)	1.000 (0.000)	1.000 (0.000)	0.738 (0.179)	0.595 (0.104)
particle	0.886 (0.116)	0.931 (0.090)	0.555 (0.030)	0.976 (0.051)	0.985 (0.039)	0.506 (0.006)	0.838 (0.135)	0.901 (0.112)	0.798 (0.140)	0.542 (0.054)
Average Ranking	5.24	4.93	7.08	3.82	2.98	9.37	5.57	5.34	5.11	5.56
Nemenyi p -value	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	-	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$

References

- [Alippi et al., 2017] Alippi, C., Boracchi, G., and Carrera, D. (2017). Ccm: Controlling the change magnitude in high dimensional data. In Angelov, P., Manolopoulos, Y., Iliadis, L., Roy, A., and Vellasco, M., editors, *Advances in Big Data*, pages 216–225, Cham. Springer International Publishing.
- [Balakrishnan and Rao, 1998] Balakrishnan, N. and Rao, C. R. (1998). Handbook of statistics. v. 16: Order statistics: theory and methods.
- [Boracchi et al., 2018] Boracchi, G., Carrera, D., Cervellera, C., and Maccio, D. (2018). Quanttree: Histograms for change detection in multivariate data streams. In *International Conference on Machine Learning*, pages 639–648. PMLR.
- [Dal Pozzolo et al., 2017] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., and Bontempi, G. (2017). Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE transactions on neural networks and learning systems*, 29(8):3784–3797.
- [Dua and Graff, 2017] Dua, D. and Graff, C. (2017). UCI machine learning repository.
- [Lehmann et al., 2005] Lehmann, E. L., Romano, J. P., and Casella, G. (2005). *Testing statistical hypotheses*, volume 3. Springer.
- [Papoulis and Pillai, 2002] Papoulis, A. and Pillai, S. U. (2002). *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education.
- [Shapiro and Wilk, 1965] Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.
- [Souza et al., 2020] Souza, V. M., dos Reis, D. M., Maletzke, A. G., and Batista, G. E. (2020). Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*, 34(6):1805–1858.
- [Tipping, 1999] Tipping, M. E. (1999). Deriving cluster analytic distance functions from gaussian mixture models.