Supplementary Materials

Thresholds

In Table 1 we report the polynomials (in $1/t$) approximating the thresholds of QT-EWMA for different target ARL_0.

Table 1. Polynomial coefficients.

target ARL_0	500	1000	2000	5000	10000	20000
constant	0.814	0.878	0.940	1.01	1.07	1.13
$\times t^{-1}$	2.53	2.51	1.88	1.81	2.10	2.41
$\times t^{-2}$	$-2.96 \cdot 10^2$	$-3.09 \cdot 10^2$	$-2.70 \cdot 10^2$	$-2.74 \cdot 10^2$	$-3.07 \cdot 10^2$	$-3.52 \cdot 10^2$
$\times t^{-3}$	$4.93 \cdot 10^3$	$5.13 \cdot 10^3$	$3.83 \cdot 10^3$	$3.75 \cdot 10^3$	$4.51 \cdot 10^3$	$6.11 \cdot 10^3$
$\times t^{-4}$	$-4.04 \cdot 10^4$	$-4.12 \cdot 10^4$	$-2.33 \cdot 10^4$	$-2.11 \cdot 10^4$	$-2.93 \cdot 10^4$	$-5.56 \cdot 10^4$
$\times t^{-5}$	$1.87 \cdot 10^5$	$1.83 \cdot 10^5$	$5.71 \cdot 10^4$	$3.96 \cdot 10^4$	$8.62 \cdot 10^4$	$3.04 \cdot 10^5$
$\times t^{-6}$	$-5.02 \cdot 10^5$	$-4.57 \cdot 10^5$	$2.37 \cdot 10^4$	$8.97 \cdot 10^5$	$-5.79 \cdot 10^5$	$-1.01 \cdot 10^6$
$\times t^{-7}$	$7.61 \cdot 10^5$	$6.22 \cdot 10^5$	$-3.78 \cdot 10^5$	$-5.05 \cdot 10^5$	$-2.46 \cdot 10^5$	$1.96 \cdot 10^6$
$\times t^{-8}$	$-5.95 \cdot 10^5$	$-4.18 \cdot 10^5$	$6.25 \cdot 10^5$	$7.46 \cdot 10^5$	$5.12 \cdot 10^5$	$-1.98 \cdot 10^6$
$\times t^{-9}$	$1.85 \cdot 10^5$	$1.06 \cdot 10^5$	$-3.08 \cdot 10^5$	$-3.52 \cdot 10^5$	$-2.69 \cdot 10^5$	$7.79 \cdot 10^5$

Proof of Propositions 1 and 2

Proposition 1. *Let the threshold h^ν be such that*

$$\mathbb{P}(T^\nu(W) > h^\nu) = \alpha, \tag{1}$$

where W is a batch of ν samples drawn from ϕ_0. Then, the monitoring scheme $T^\nu(W) > h^\nu$ yields $ARL_0 \geq \nu/\alpha$.

Proof. Let t_B^* the first time instant such that $T^\nu(W_{t_B^*}) > h$ and let us compute $\mathrm{ARL}_0 = \mathbb{E}[t_B^*]$. To this purpose, we follow a strategy similar to that in Section 4.2. At first we observe that since the batches does not overlap, the variables $\{T^\nu(W_t)\}$ are independent if we condition w.r.t. the specific training set realization (thus the model used to compute $\mathcal{T}^\nu$). Therefore, we obtain that:

$$\mathbb{P}(T^\nu(W_t) > h \mid TR, T^\nu(W_k) \leq h \; \forall k < t) = \mathbb{P}(T^\nu(W_t) > h \mid TR). \tag{2}$$

Let us define the random variable $p = \mathbb{P}(T^\nu(W) > h \mid TR)$, where W is a batch of ν samples drawn from ϕ_0. Following [3], the random variable t_B^* is distributed as a geometric random variable w.r.t. the conditional probability $\mathbb{P}(\cdot|TR)$, and its expected value is

$$\mathbb{E}[t_B^* \mid TR] = \frac{1}{p}. \tag{3}$$

To compute the ARL_0 we only have to evaluate the expectation of $1/p$ w.r.t. to the training set realizations since the law of total expectation implies that

$$\mathbb{E}[t_B^*] = \mathbb{E}[\mathbb{E}[t_B^* \mid TR]] = \mathbb{E}\left[\frac{1}{p}\right]. \tag{4}$$

We observe that Jensen's inequality implies that

$$\mathbb{E}[t_B^*] = \mathbb{E}\left[\frac{1}{p}\right] \geq \frac{1}{\mathbb{E}[p]}, \tag{5}$$

since the function $1/p$ is convex for $p > 0$. Finally, we have to compute $\mathbb{E}[p]$. To this purpose we rewrite

$$p = \mathbb{P}(T^\nu(W) > h^\nu \mid TR) = \mathbb{E}[\mathbb{1}(T^\nu(W) > h^\nu) \mid TR], \tag{6}$$

where $\mathbb{1}$ denotes the indicator function. Then:

$$\mathbb{E}[p] = \mathbb{E}[\mathbb{E}[\mathbb{1}(T^\nu(W) > h^\nu)|TR]] = \tag{7}$$

$$= \mathbb{E}[\mathbb{1}(\{T^\nu(W) > h^\nu\})] = \mathbb{P}(T^\nu(W_t) > h^\nu) = \alpha, \tag{8}$$

where the equality between (7) and (8) is due to the law of total expectation. Combining (7) and (5) we obtain that

$$\mathbb{E}[t_B^*] \geq \frac{1}{\alpha}. \tag{9}$$

To obtain the thesis we observe that, since the monitoring is performed in a batch-wise manner, change detected after the t_B^* batch translates in a detection made after $\nu \cdot t_B^*$ samples of the datastream, so $\mathrm{ARL}_0 \geq \nu/\alpha$.

Proposition 2. *Let the threshold h^ν be such that*

$$\mathbb{P}(T^\nu(W) > h^\nu \mid TR) = \alpha, \tag{10}$$

where W is a batch of ν samples drawn from ϕ_0. Then, the monitoring scheme $T^\nu(W) > h^\nu$ yields $ARL_0 = \nu/\alpha$.

Proof. Following the same strategy we pursued to prove Proposition 1, we have that the random variable $p = \mathbb{P}(T^\nu(W) > h^\nu|TR)$ is a constant equal to α. Therefore, the equality holds in (5), from which we derive $\mathrm{ARL}_0 = \nu/\alpha$.

Additional experiments

Empirical $\mathbf{ARL_0}$. The comparison of empirical - target ARL_0 on simulated Gaussian datastreams are reported in Figure 1(a,c,e), which show that QT-EWMA and SPLL-CPM control the ARL_0 very accurately, regardless of the data dimension, while the empirical ARL_0 of QT is higher than the target, as we expected. Figures 3(a), 4(a,c,e), and 5(a,c,e) show that we obtain the same result on datastreams sampled from the considered real-world dataset, confirming the nonparametric nature of QuantTree. In contrast, SPLL and Scan-B cannot maintain a high target ARL_0, both on simulated and real-world datastreams due to an inaccurate estimate of the detection thresholds.

Detection delay vs false alarms. We plot the percentage of false alarms against the average detection delay, setting different ARL_0 values, to assess the

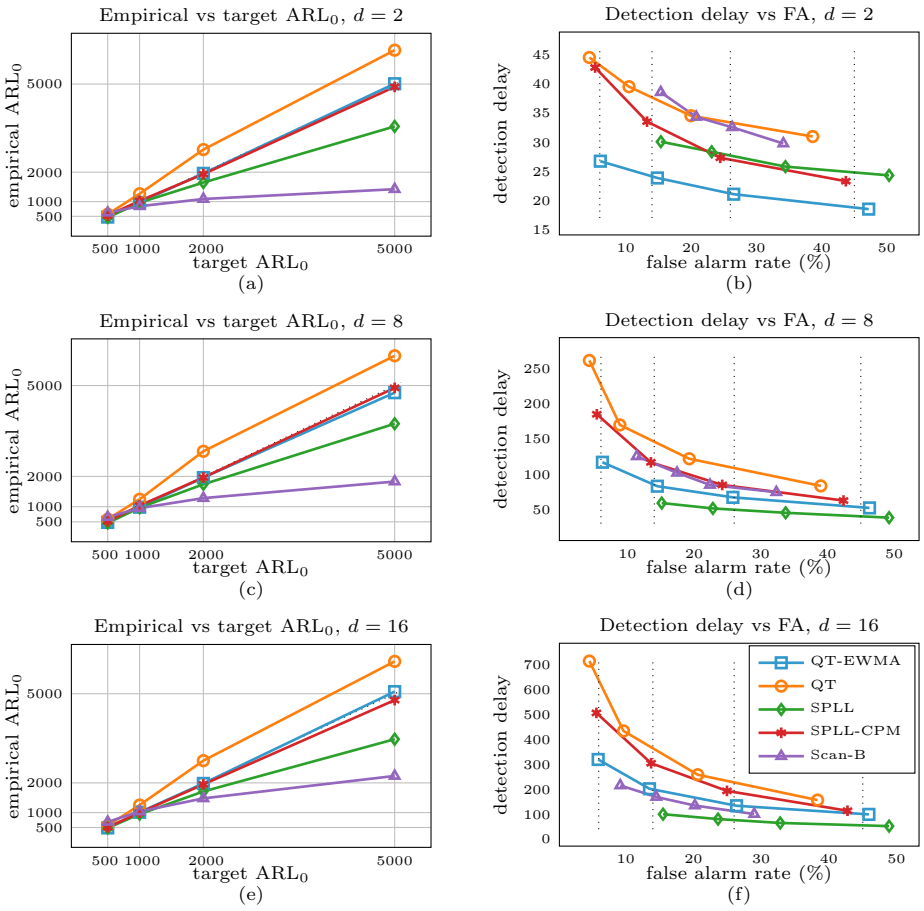

Fig. 1. Experimental results obtained on d-dimensional Gaussian datastreams ($d \in \{2, 8, 16\}$). (a,c,e) show the empirical ARL$_0$ of the considered methods. The target ARL$_0$ $\in \{500, 1000, 2000, 5000\}$ is maintained when the line is close to the dotted diagonal. (b,d,f) show the average detection delay of the considered methods plotted against the percentage of false alarms, which should approach the dotted false alarm rates when the target ARL$_0$ is maintained.

trade-off between these two quantities. Figures 1(b,d,f) show the performance of the considered methods on simulated Gaussian datastreams ($d \in \{2, 8, 16\}$, respectively) containing a change point at $\tau = 300$ such that $\mathrm{sKL}(\phi_0, \phi_1) = 2$. In terms of detection delay, QT-EWMA is the best method when $d = 2$, while SPLL outperforms all the other methods when $d \in \{8, 16\}$, which is expected since its parametric assumptions are met (ϕ_0 is a Gaussian). All methods decrease their power as the data dimension d increases, which is an expected effect of the *detectability loss*. Scan-B is the best method on the credit dataset [1]

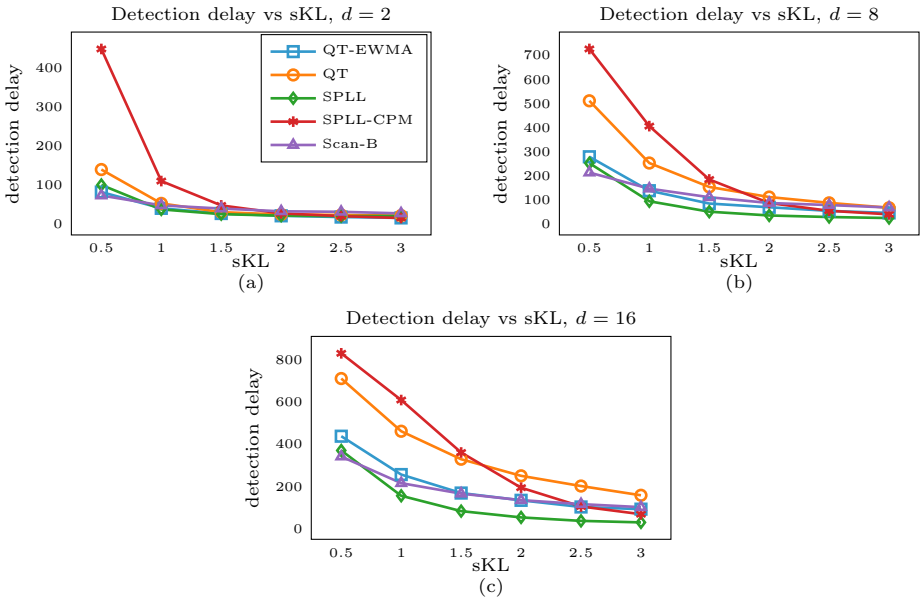

Fig. 2. Detection delay as a function of the change magnitude computed on simulated Gaussian datastreams with dimension $d \in \{2, 8, 16\}$ containing a change point at $\tau = 300$ with target $\text{ARL}_0 = 1000$ (which is maintained by all methods, see Figure 1)

(Figure 3(b)), while QT-EWMA achieves lower detection delays on all the UCI datasets [2], independently from d. The performance of Scan-B is particularly poor compared to the other methods on the sensorless, spruce and lodgepole datasets (Figures 4(b) and 5(d,f)). Remarkably, the performance of SPLL-CPM strongly depends on the data: it is the quickest method in detecting changes on the sensorless, spruce and lodgepole datasets (Figures 4(b) and 5(d,f)), while its delays are higher compared to the other methods on credit, particle, protein and niño (Figures 3(b), 4(d,f) and 5(b)). We speculate that this is due to an imperfect fit of the GMM on these datasets. The fact that QT-EWMA consistently outperforms QT indicates that our sequential statistic is more sensitive to distribution changes in streaming data than QuantTree statistic designed for batch-wise monitoring. In terms of false alarms, QT-EWMA and SPLL-CPM approach the target values, while QT and SPLL have, respectively, fewer and more false alarms, as a consequence of their empirical ARL_0, and this happens in all the considered monitoring scenarios. The false alarms of Scan-B, instead, exhibit a completely different behaviour, which also depends on the data distribution. This is due to the fact that its thresholds are not designed to yield a constant false alarm probability.

Detection delay. Figure 2 shows the detection delays of the considered methods on Gaussian datastreams containing as a function of the change magnitude. To enable a fair comparison, we configured all methods by setting the target

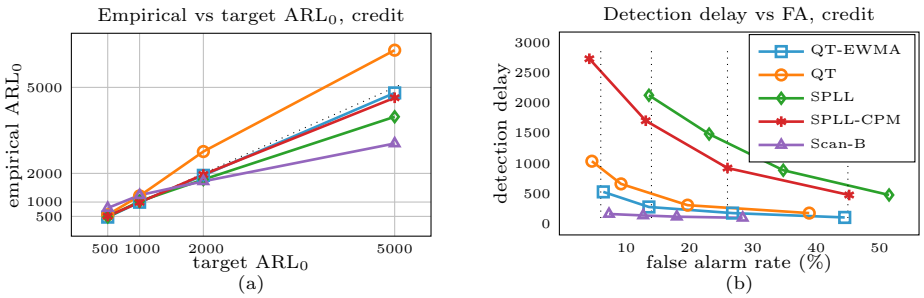

Fig. 3. Experimental results obtained on datastreams sampled from the credit dataset ($d = 28$). (a) shows the empirical ARL_0 of the considered methods configured with target $\text{ARL}_0 \in \{500, 1000, 2000, 5000\}$, which is maintained when the line is close to the dotted diagonal. (b) shows the detection delay of the considered methods plotted against the percentage of false alarms, which should approach the dotted false alarm rates when the target ARL_0 is maintained.

$\text{ARL}_0 = 1000$, a value that can be maintained well by all methods, as shown in Figures 1(a,c). The best method on simulated Gaussian datastreams is SPLL, which is expected to achieve excellent results since its parametric assumptions are met. The best nonparametric method is QT-EWMA. As observed also in Figure 1(b,d). As expected, all methods decrease their detection delays when the change magnitude increases.

References

1. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems **29**(8), 3784–3797 (2017)
2. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
3. Margavio, T.M., Conerly, M.D., Woodall, W.H., Drake, L.G.: Alarm rates for quality control charts. Statistics & Probability Letters **24**(3), 219–224 (1995)

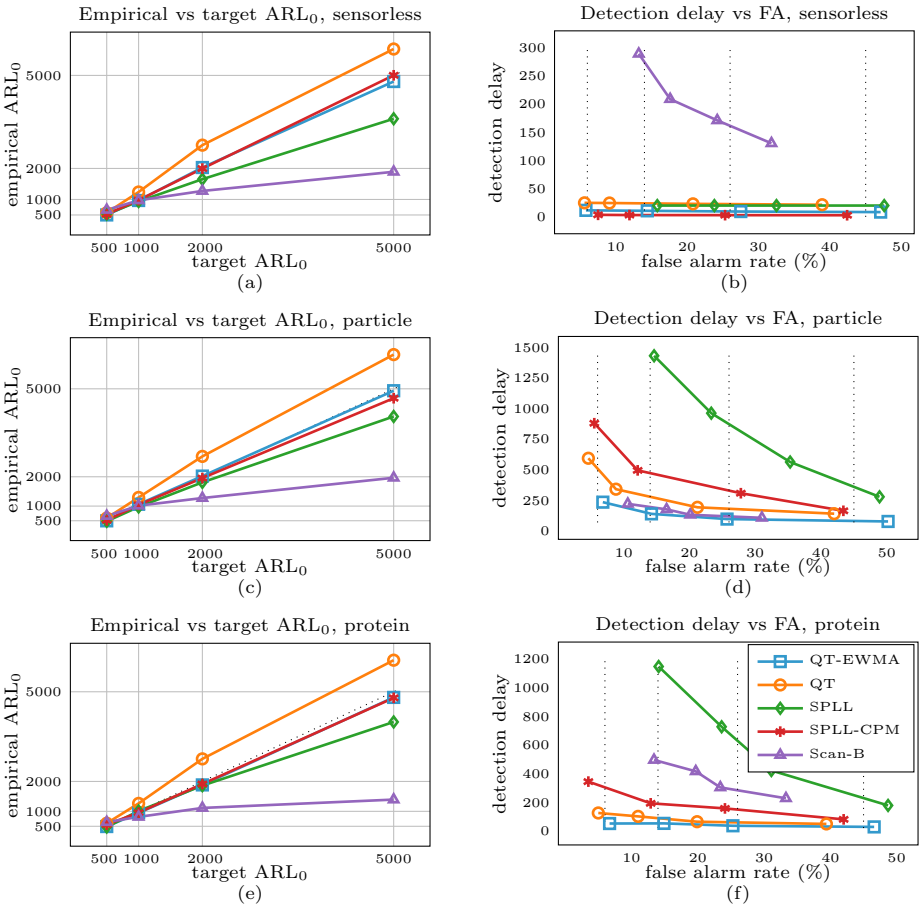

Fig. 4. Experimental results obtained on datastreams sampled from UCI datasets sensorless ($d = 50$), particle ($d = 48$), and protein ($d = 9$). (a,c,e) show the empirical ARL_0 of the considered methods configured with target $\mathrm{ARL}_0 \in \{500, 1000, 2000, 5000\}$, which is maintained when the line is close to the dotted diagonal. (b,d,f) show the detection delay of the considered methods plotted against the percentage of false alarms, which should approach the dotted false alarm rates when the target ARL_0 is maintained.

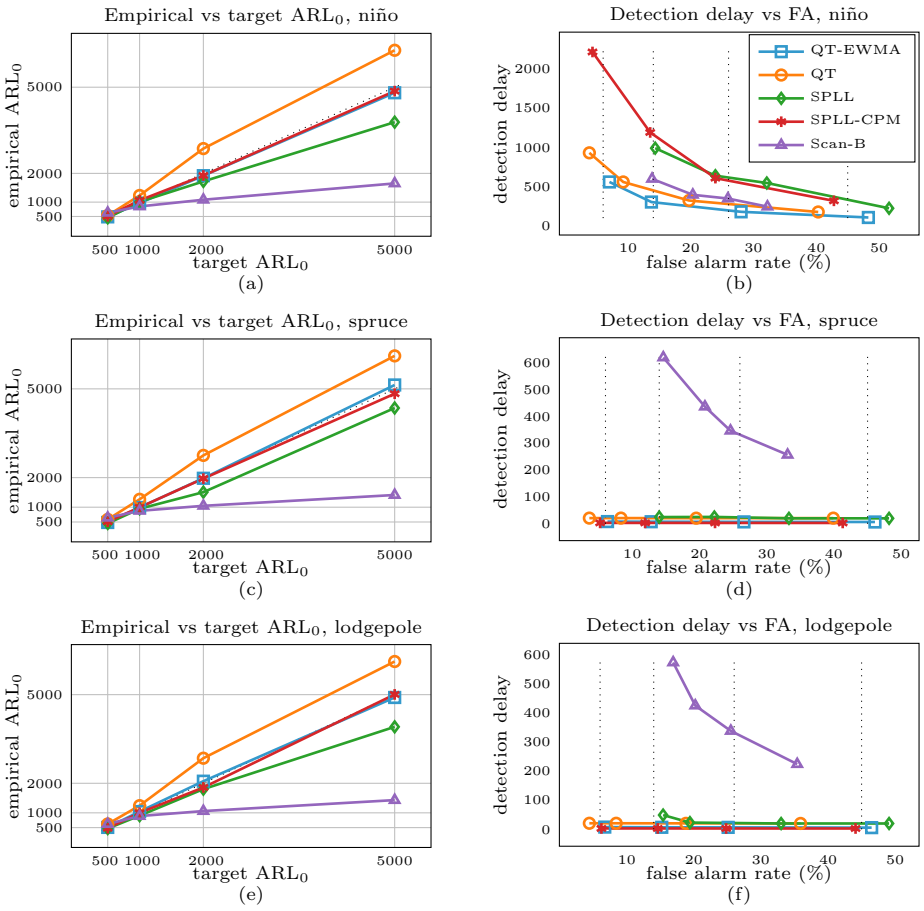

Fig. 5. Experimental results obtained on datastreams sampled from UCI datasets niño ($d = 5$), spruce ($d = 10$), and lodgepole ($d = 10$). (a,c,e) show the empirical ARL_0 of the considered methods configured with target $\mathrm{ARL}_0 \in \{500, 1000, 2000, 5000\}$, which is maintained when the line is close to the dotted diagonal. (b,d,f) show the detection delay of the considered methods plotted against the percentage of false alarms, which should approach the dotted false alarm rates when the target ARL_0 is maintained.