Supplemental material

Daniele Zago and Giovanna Capizzi

2022-09-28

Here, we implement the code that reproduces the analysis on the ICU dataset in Section 5 of the paper.

First, we load the necessary packages and the environment variables that are provided in the source code.

```
using DrWatson
#@quickactivate "CautiousLearning"
quickactivate("$(homedir())/Documents/git/SPC/CautiousLearning")

using Distributions, Random
using Parameters
using SharedArrays
using DataFrames, CSV, Dates
using Plots, StatsBase, StatsPlots, LaTeXStrings, RCall
using StatisticalProcessControl

include(srcdir("generate_data.jl"))
include(srcdir("update_parameter.jl"))
include(srcdir("simulate_runs.jl"))
```

```
Error: LoadError: UndefVarError: SimulationSettings not defined
in expression starting at /home/dede/Documents/git/SPC/CautiousLearning/src
/simulate_runs.jl:206
```

First, we load the dataset which is located in the `data/ICUadmissions` folder, relative to the main directory of the project. We consider the data concerning the New York City area in 2020.

```
fold = "ICUadmissions"
dat = DataFrame(CSV.File(datadir(fold,
"New_York_Forward_COVID-19_Daily_Hospitalization_Summary_by_Region.csv")))
nydat = filter(row -> row.Region == "NEW YORK CITY", dat)
nydat.date .= Date.(nydat[:, 1], dateformat"mm/dd/yyyy")
nydat = filter(row -> year(row.date) == 2020, nydat);
```

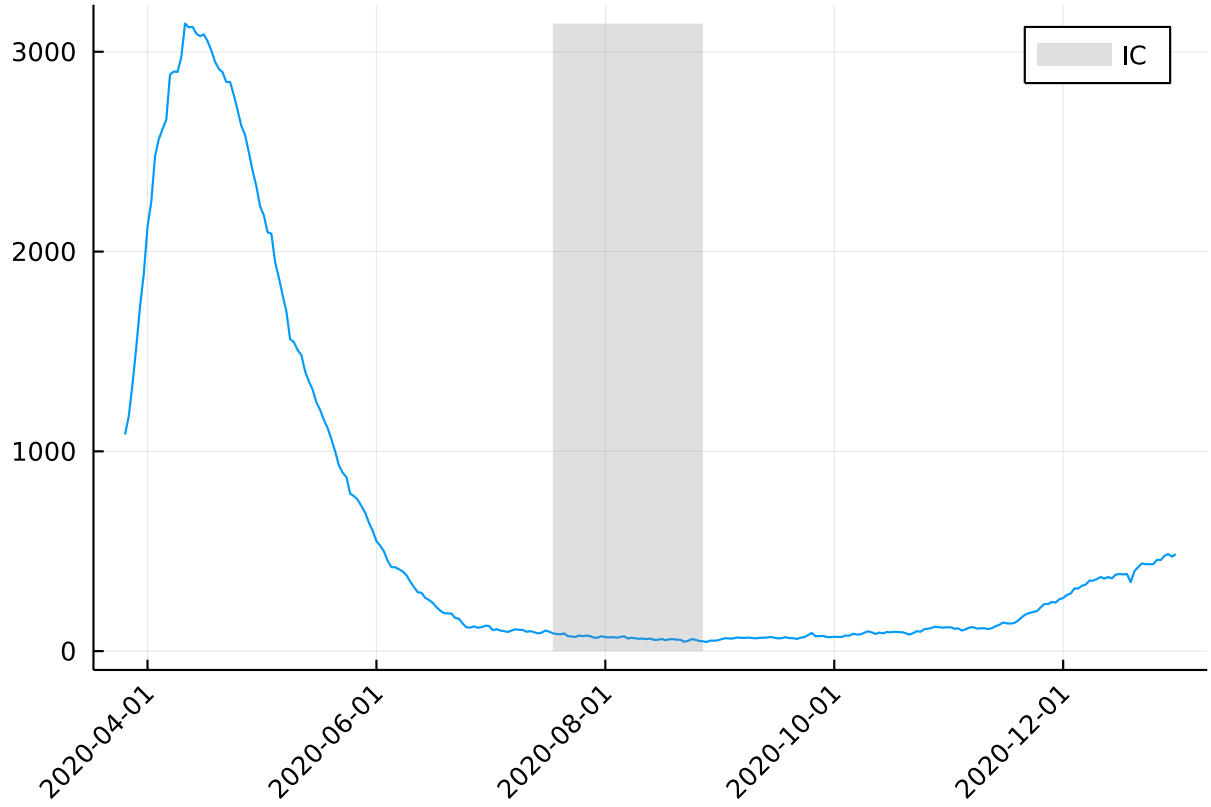
We obtain the daily ICU counts along with the corresponding dates, which start from March 2020.

```
using Plots.PlotMeasures
y2020 = nydat[:, 4]
days2020 = nydat[:, 5]
first(y2020, 10)
first(days2020, 10)
```

```
10-element Vector{Dates.Date}:
 2020-03-26
 2020-03-27
 2020-03-28
 2020-03-29
 2020-03-30
 2020-03-31
 2020-04-01
 2020-04-02
 2020-04-03
 2020-04-04
```

Then, we select the indices of the IC/OC datasets along with the initial sample size and prospective monitoring size. The following code reproduces Figure ?? in the paper.

```
# ic_2020 = 115:175
ic_2020 = 115:155
oc_2020 = (ic_2020[end]+1):length(y2020)
n_ic = length(ic_2020)
n_oc = length(oc_2020)
pl = plot(days2020, y2020, label="", dpi=400, xrotation=45, bottom_margin=3mm)
plot!(pl, days2020[ic_2020], fill(0, n_ic), fillrange=fill(maximum(y2020), n_ic),
color=:gray, fillcolor = "gray", fillalpha=0.25, alpha=0.0, label="IC")
# safesave(plotsdir(fold, "ICU-cases-2020"),pl)
```

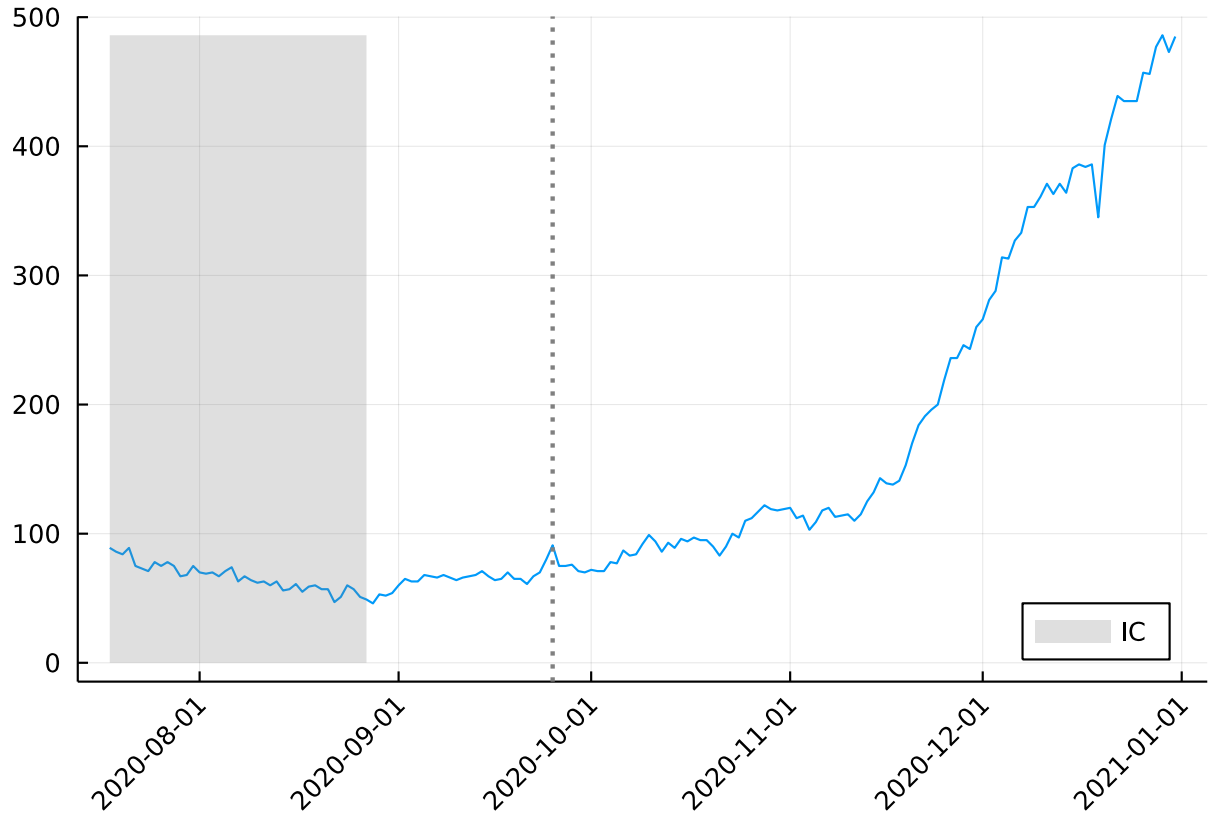


We then isolate the IC and OC data. This code reproduces Figure ?? in the paper.

```
y = y2020[[ic_2020; oc_2020]]
days = days2020[[ic_2020; oc_2020]]

ic_idx = 1:n_ic
oc_idx = (n_ic+1):(n_ic+n_oc)
yIC = y[ic_idx]
```

```
daysIC = days[ic_idx]
yOC = y[oc_idx]
daysOC = days[oc_idx]

println("In-control data: ", daysIC[[1, end]])
pl = plot(days, y, label="", dpi=400, xrotation=45, bottom_margin=3mm)
plot!(pl, days[1:n_ic], fill(0, n_ic), fillrange=fill(maximum(y), n_ic), color=:gray,
fillcolor = "gray", fillalpha=0.25, alpha=0.0, label="IC", legend=:bottomright)
τ = n_ic + 29
vline!([days[τ]], color=:gray, linestyle=:dot, linewidth=2,  label="", markersize=2.5)
display(pl)
# safesave(plotsdir(fold, "ICU-IC-OC.png"), pl)

println("Possible change-point location: ", days[τ])
```

```
In-control data: [Dates.Date("2020-07-18"), Dates.Date("2020-08-27")]
Possible change-point location: 2020-09-25
```



We then calculate the initial estimate $\hat{\theta}$ and set the nominal IC ARL alongside the GICP probability β.

```
thetaHat = mean(yIC)
Arl0 = 500
beta = 0.05

0.05
```

Next, we define a function that implements the control chart procedures using the various update mechanisms (AE, FE, CLM) that are defined in the source code.

```
function applyChart(ch, um, thetaHat, m, yprosp; seed = 123)
    # Random.seed!(seed)
    maxrl_i = length(yprosp)
```

```
    thetaHatVec = zeros(maxrl_i)
    thetaHatVec[1] = thetaHat
    di = 1
    diVec = Array{Int}(undef, maxrl_i)
    diVec[1] = di
    thetaHatCaut = thetaHat
    thetaHatCautVec = zeros(maxrl_i)
    thetaHatCautVec[1] = thetaHatCaut

    t_alarm = zeros(0)

    valueVec = zeros(maxrl_i)
    valueVec[1] = get_value(ch)
    i = 1
    while i < maxrl_i
        thetaHatCaut = thetaHatVec[i - di + 1]
        y = yprosp[i]
        ch = update_series(ch, chart_statistic(y, thetaHatCaut))
        thetaHat = update_parameter(thetaHat, y, i + m)
        i += 1
        valueVec[i] = get_value(ch)
        thetaHatVec[i] = thetaHat
        if check_update(ch, um)
            di = 1
        else
            di += 1
        end
        diVec[i] = di
        thetaHatCautVec[i] = thetaHatCaut
        if check_OC(ch)
            push!(t_alarm, i)
        end
    end

    return (t_alarm = t_alarm, dat = yprosp, chart_values = valueVec, limit_alarm =
get_limits(ch), limit_cautious = get_warning_limit(um), parameter_updates =
thetaHatCautVec, di = diVec)
end

applyChart (generic function with 1 method)
```

Furthermore, we define a function that applies the control chart to a prospective monitoring data after having corrected the control limit to satisfy the GICP condition.

```
function applyChartGICP(ch, um, yinit, yprosp, thetaHat, Arl0; beta::Union{Bool,
Float64} = 0.2, maxrl=1e04, verbose=true, seed=Int(rand(1:1e06)))
    m = length(yinit)
    if isa(um, CautiousLearning)
        Ats0 = get_ATS(um)
        if Ats0 != 0
            # Calculate limit if Ats0 != 0, otherwise use zero-restarting chart
            if verbose println("Calculating limits for target ATS...") end
            sa_ats = saControlLimits(ch, AdaptiveEstimator(), runSimulation, Ats0,
thetaHat, Poisson(thetaHat),
                                     m, verbose=false, Amin=0.1, maxiter=1e05,
                                     gamma=0.015, adjusted=true, seed=seed)
            um = CautiousLearning(L = sa_ats[:h], ATS = Ats0)
        else
            if verbose println("ATS = 0, skipping limit calculation.") end
        end
```

```
        # Estimate cautious learning limit
    end

    if beta == false
        chart = deepcopy(ch)
    else
        chart = adjust_chart_gicp(ch, um, yinit, thetaHat, runSimulation, m, Arl0,
beta=beta, verbose=verbose)
    end

    return applyChart(chart, um, thetaHat, m, yprosp)
end

applyChartGICP (generic function with 1 method)
```

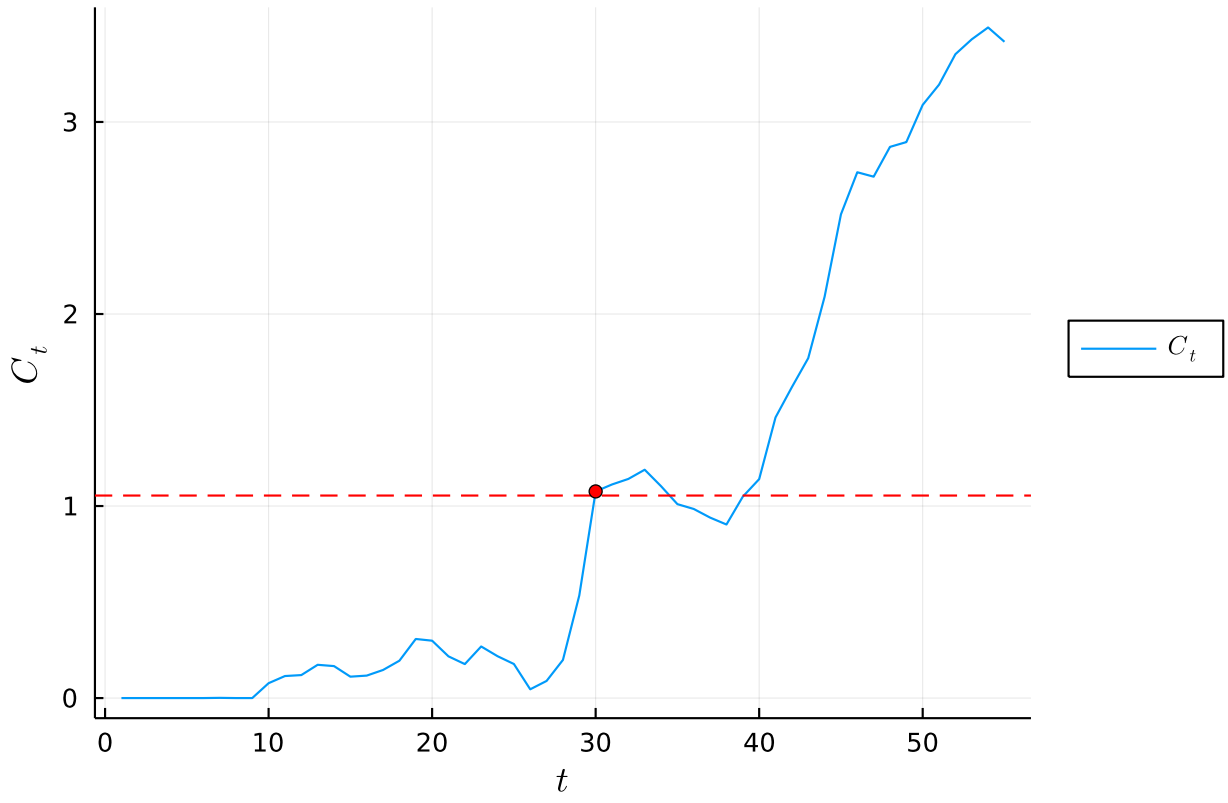
The GICP control limit correction uses the `adjust_chart_gicp` function defined in the source code, which implements the computations described in Subsection ??.

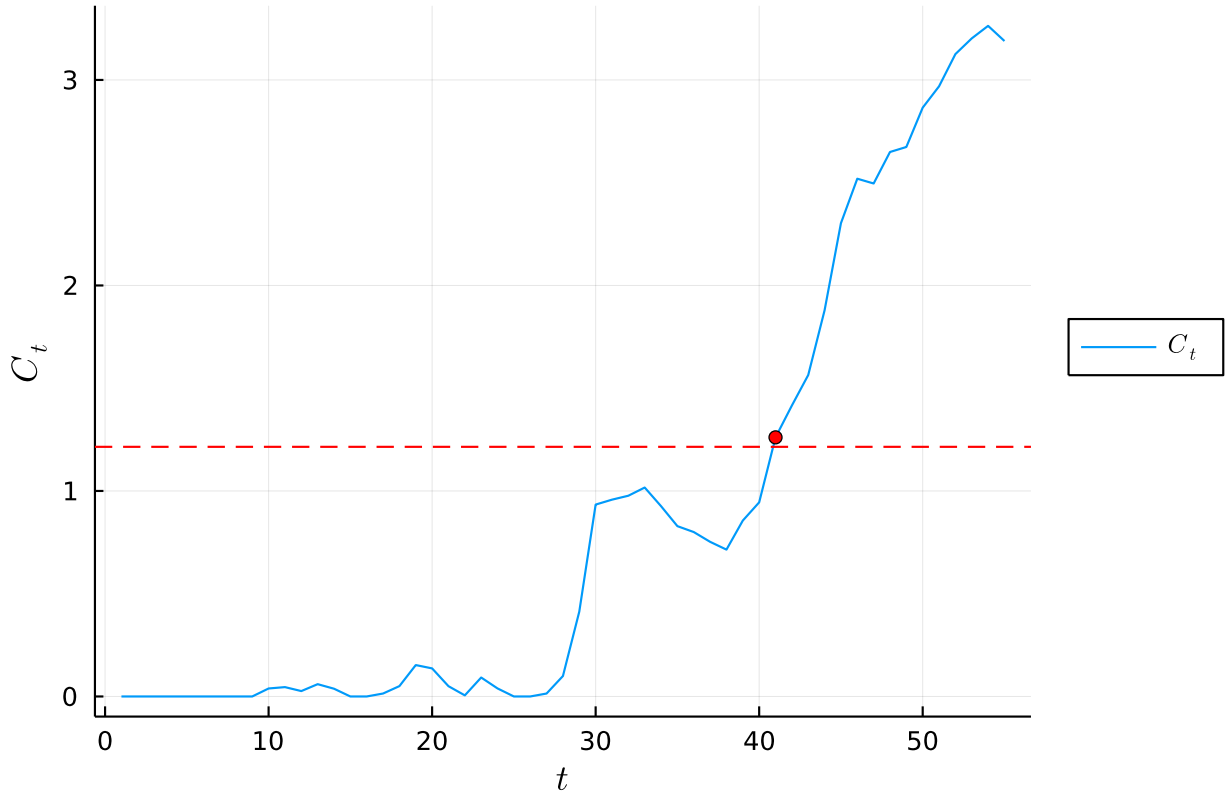
We finally apply the control charts using the methods and obtain the results. This code reproduces Figure ?? and Table ??.

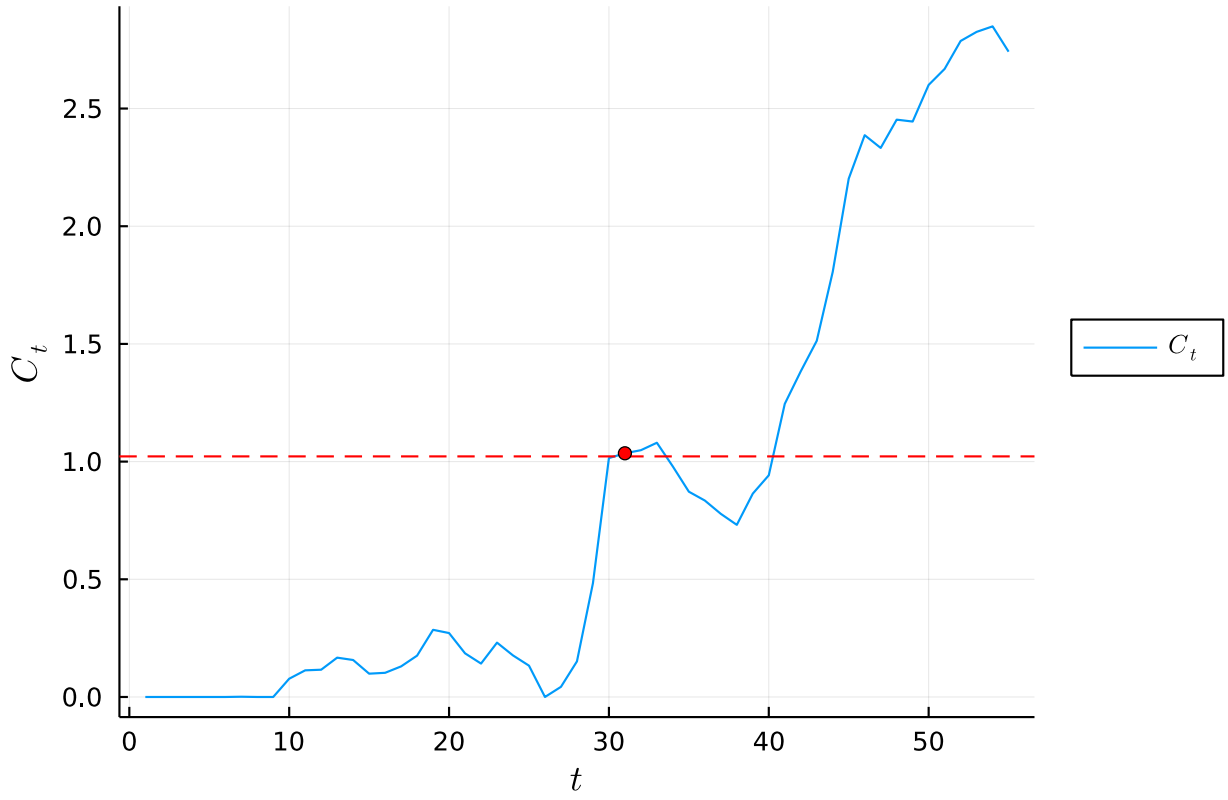
```
Random.seed!(2022-08-18)
D = Poisson
fname = plotsdir(fold, "alarms.jld2")

umVec = [CautiousLearning(ATS=0), FixedParameter(), AdaptiveEstimator()]
nms = ["CLM", "FE", "AE"]
perf = []
L = []
for i in eachindex(umVec)
    ch = signedEWMA(l=0.2, L = 1.0)
    um = umVec[i]
    res = applyChartGICP(ch, um, yIC, yOC, thetaHat, Arl0, beta=beta)
    append!(L, res[:limit_alarm])
    plotsave = plotsdir(fold, nms[i]*".png")
    pl = plot(res.chart_values[1:55], label=L"C_t", legend=:outerright, dpi=400)
    hline!([res.limit_alarm], style=:dash, colour="red", xlab=L"t", ylab=L"C_t",
label="")
    tau = Int(first(res.t_alarm))
    scatter!([tau], [res.chart_values[tau]], colour="red", label="")
    display(pl)
    append!(perf, tau)
    println(tau,"\t", daysOC[tau])
end

@rput nms
@rput perf
@rput L

tab = R"""
library(knitr)
library(kableExtra)
df = cbind(nms, perf, L)
tex_OC <- kable(df, format="latex", booktabs=TRUE, digits = 2, row.names=FALSE,
col.names=c("Estimator", "Alarm", "Limit"), escape=FALSE, align='c', linesep = "",
    caption = "Time to alarm of the one-sided EWMA control chart using the fixed (FE),
adaptive (AE) and the proposed cautious learning (CLM) update rules.", label="ICU OC
alarm") %>%
    kable_styling(latex_options = "HOLD_position")
""" |> rcopy
```

```
println(tab)
```

```
ATS = 0, skipping limit calculation.
Calculating GICP for extremum 1/1...
GICP done.
30      2020-09-26
Calculating GICP for extremum 1/1...
GICP done.
41      2020-10-07
Calculating GICP for extremum 1/1...
GICP done.
31      2020-09-27
\begin{table}[H]

\caption{\label{tab:ICU OC alarm}Time to alarm of the one-sided EWMA contro
l chart using the fixed (FE), adaptive (AE) and the proposed cautious learn
ing (CLM) update rules.}
\centering
\begin{tabular}[t]{ccc}
\toprule
Estimator & Alarm & Limit\\
\midrule
CLM & 30 & 1.05469060798468\\
FE & 41 & 1.2147825075143\\
AE & 31 & 1.02214273823991\\
\bottomrule
\end{tabular}
\end{table}
```




C_t
3
2
1
0
0
10
20
30
40
50
t
C_t


C_t
2.5
2.0
1.5
1.0
0.5
0.0
0
10
20
30
40
50
t
C_t