4.4. Non-parametric analysis

An E-divisive method was applied to all three datasets. The method reported one cluster for the dataset of the software release L16A. There were five clusters found in both datasets of the software release L16B and L17A. Table 4.5 shows places in the time series data where the E-divisive method is able to detect the significant changes.

Table 4.5.: The locations of the statistically significant change points from applying the E-divisive algorithm in each dataset

Software release	Change-point location
L16A	-
L16B	130, 135, 153, 170
L17A	9, 77, 82, 105

The CPU utilization of the software release L16A, L16B and L17A along with its estimated change points in the time series are plotted and shown later in sec. 4.5.3.

4.5. Comparison between the Markov switching model and the E-divisive method

A comparison between the Markov switching model and the E-divisive method was made in this section. Two methods were applied to two simulated datasets, where the true changes are already known beforehand, and a real data.

4.5.1. Simulated Dataset 1

In Figure 4.13, there are nine estimated change point locations in the simulated data. The Markov switching model detects more changes than what the data actually has. In contrast, the E-divisive method is able to identify changes less than the true changes in the data, and clearly less than the results from the Markov switching model. For the Markov switching model, the model reported two more locations. The plot illustrates that the model performs rather well in detecting the changes in the data. The model discovers most changes when they occur. However, there is one location where the model detect the change later than the actual change.

A delay in detecting the changes for the E-divisive method can be seen in the plot. In total, there are three change point locations where the method identifies changes later than the true changes. Furthermore, one location is discovered by this method before the actual change occurs. The E-divisive method is unable to detect any changes in the data at the beginning of the time period.

When these two methods detect changes after the true changes, most of their estimated change point locations are only behind one or two time index. To sum up, from this dataset, the Markov switching model had more false alarms while the E-divisive had more missed detections.

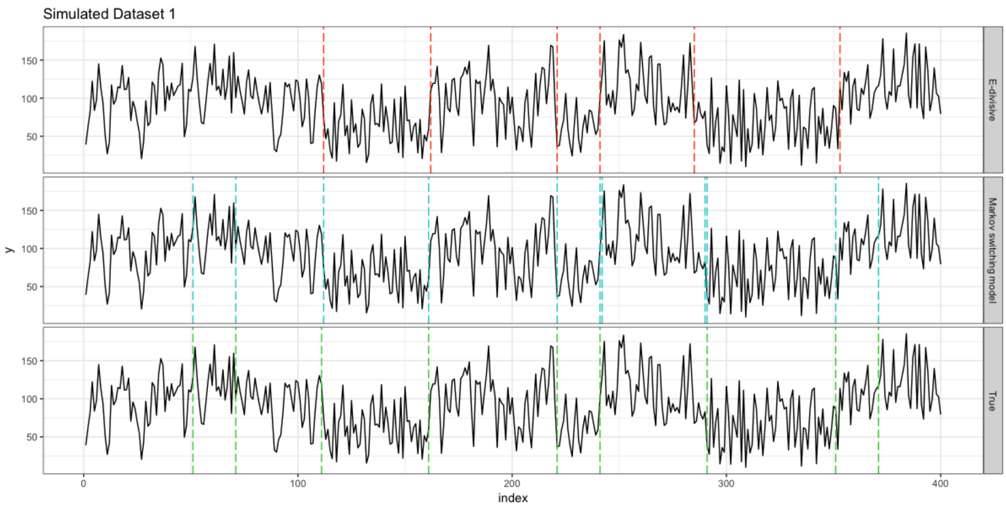

Figure 4.13.: The simulated Dataset 1 showing the estimated change point locations indicated by dashed vertical lines from the E-divisive method (Top) and the Markov switching model (Middle). The true change points in the data are indicated in the bottom plot.

4.5.2. Simulated Dataset 2

Figure 4.14 presents estimated change point locations of the E-divisive method and the Markov switching model, and also true change points in the data. The simulated dataset clearly contains numerous switches between states. Generally, the Markov switching model is able to identify the changes considerably well despite a few false alarms and missed detections.

For the E-divisive method, two estimated change points are identified. The method is correctly estimated the location of the change in the data. However, the method has the worst performance in detecting changes as many change points are obviously failed to detect.

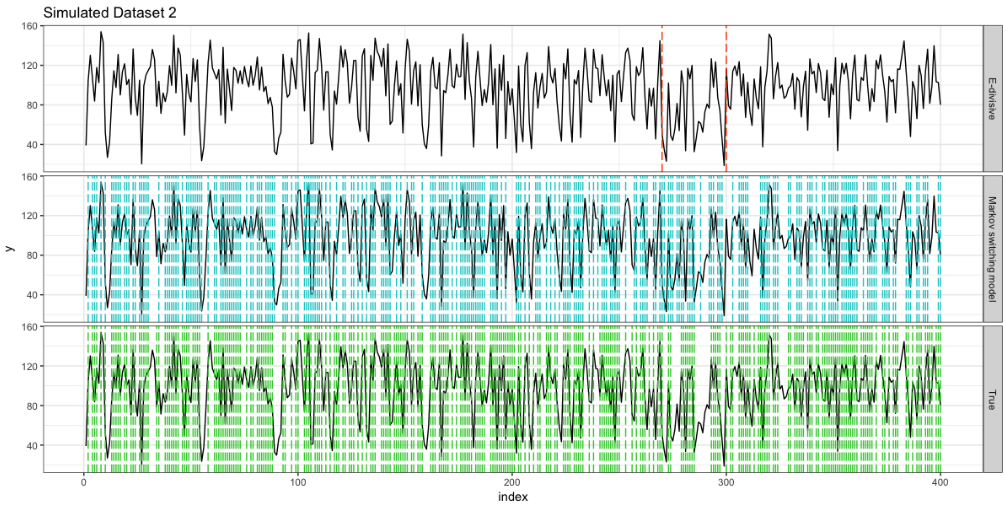

Figure 4.14.: The simulated Dataset 2 showing the estimated change point locations indicated by dashed vertical lines from the E-divisive method (Top) and the Markov switching model (Middle). The true change points in the data are indicated in the bottom plot.

4.5.3. Real data

4.5.3.1. Software release L16A

According to Table 4.5, the E-divisive method could not identify any changes in the dataset of the software release L16A. Thus, a comparison between two methods could not be made for this dataset.

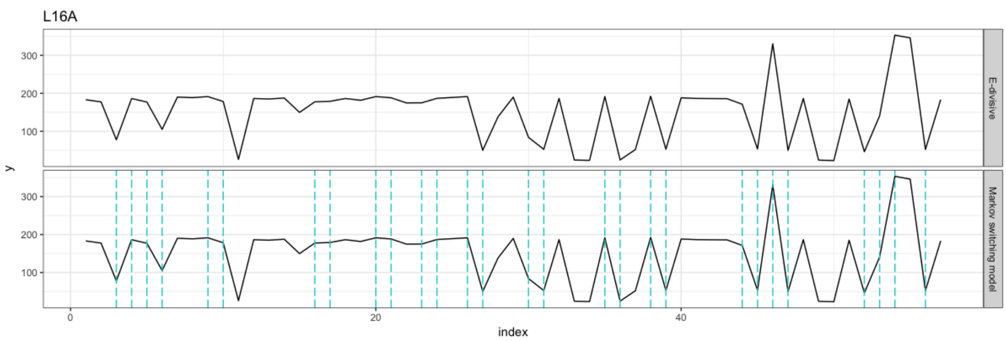

Figure 4.15.: The CPU utilization of the software release L16A. The dashed vertical lines indicate the locations of estimated change points from the Markov switching model (Bottom).

4.5.3.2. Software release L16B

Four estimated change point locations are identified from the E-divisive method for the software release L16B. They are likely to occur around the same period of time which are almost at the end of the time series data. The locations of the change point from the E-divisive method are approximately at peaks and negative peaks.

Figure 4.16 presents results of estimated change points from the Markov switching model and the E-divisive method. Many estimated change point locations are found by the Markov switching model. Apparently, two methods are able to estimate change points almost at the same time. There is even one location where both methods detect a change at the exact time. Moreover, one change point is discovered by the E-divisive method but not by the Markov switching model.

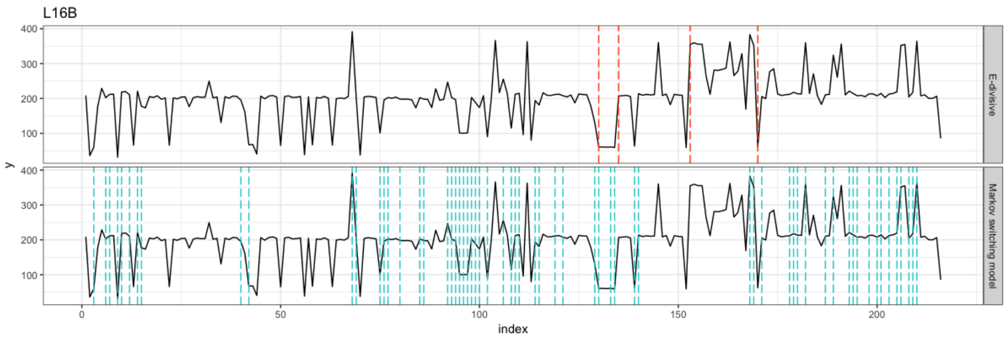

Figure 4.16.: The CPU utilization of the software release L16B. The dashed vertical lines indicate the locations of estimated change points from the E-divisive method (Top) and the Markov switching model (Bottom).

4.5.3.3. Software release L17A

There are four estimated change point locations which are identified from the E-divisive method. These locations are rather spread out. The E-divisive method discovers changes when the CPU utilization was about to drop or increase its value.

Several changes are discovered by the Markov switching model as can be seen in Figure 4.17. The two methods seem to detect changes roughly at the same time. Nevertheless, they are unable to identify the change at the exact time.

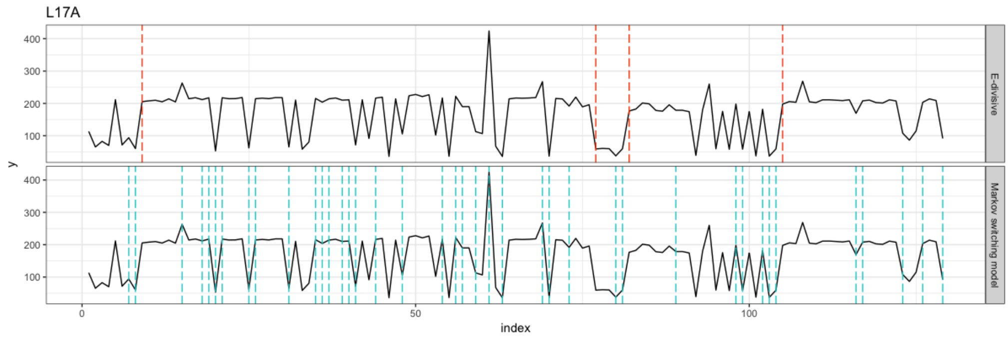

Figure 4.17.: The CPU utilization of the software release L17A. The dashed vertical lines indicate the locations of estimated change points from the E-divisive method (Top) and the Markov switching model (Bottom).

4.6. Predicting the state of the CPU utilization

In this section, the state prediction function was implemented to the test set in order to predict the most probable state for new observations.

4.6.1. Software release L16A

For the software release L16A, there are 7 observations in total. In Figure 4.18, only two states, State1 and State2, are assigned for these observations. The first three observations are in State2. Afterwards, observation tends to switch back and forth between both states until the end of the time period. Note that the most likely state for the last observation of the test set is unable to be predicted, and so it does not belong to any state.

Figure 4.18.: The predicted state of the test set in the software release L16A