signals

MDPI

Article

Online Detection and Fuzzy Clustering of Anomalies in Non-Stationary Time Series †

Changjiang He [1,*], David S. Leslie [2] and James A. Grant [2]

[1] Computer Science, University of Roehampton, London SW15 5PJ, UK
[2] Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YW, UK; d.leslie@lancaster.ac.uk (D.S.L.); j.grant@lancaster.ac.uk (J.A.G.)
* Correspondence: changjiang.he@roehampton.ac.uk
† The data used in this study include private data from the UK's national communications network and public data from Kaggle, the collection and use of the data follow the regulation of BT Group plc.

Abstract: We consider the challenge of detecting and clustering point and collective anomalies in streaming data that exhibit significant nonlinearities and seasonal structures. The challenge is motivated by detecting problems in a communications network, where we can measure the throughput of nodes, and wish to rapidly detect anomalous traffic behaviour. Our approach is to train a neural network-based nonlinear autoregressive exogenous model on initial training data, then to use the sequential collective and point anomaly framework to identify anomalies in the residuals generated by comparing one-step-ahead predictions of the fitted model with the observations, and finally, we cluster the detected anomalies with fuzzy c-means clustering using empirical cumulative distribution functions. The autoregressive model is sufficiently general and robust such that it provides the nearly (locally) stationary residuals required by the anomaly detection procedure. The combined methods are successfully implemented to create an adaptive, robust, computational framework that can be used to cluster point and collective anomalies in streaming data. We validate the method on both data from the core of the UK's national communications network and the multivariate Skoltech anomaly benchmark and find that the proposed method succeeds in dealing with different forms of anomalies within the nonlinear signals and outperforms conventional methods for anomaly detection and clustering.

Keywords: signal processing; real time; nonlinear; SCAPA; NARX; long term

Citation: He, C.; Leslie, D.S.; Grant, J.A. Online Detection and Fuzzy Clustering of Anomalies in Non-Stationary Time Series. *Signals* **2024**, *5*, 40–59. https://doi.org/10.3390/signals5010003

Academic Editor: Xiaohua Huang

Received: 1 November 2023
Revised: 15 January 2024
Accepted: 19 January 2024
Published: 24 January 2024

1. Introduction

Real-time signal processing plays a predominant role in advanced industrial systems. Recently, the demand for high-speed signal processing for streaming data has significantly increased, especially in the fields of online surveillance, real-time communication, and intelligent control [1–5]. Online analysis of streaming data can reveal valuable timely information about a system and allow rapid interventions.

Anomalies are patterns in data that differ substantially from an expected behaviour [6–8]. There are two types of anomalies prevalent in time series data: point anomalies, where single data points do not comply with the neighbouring local pattern, e.g., a banking transaction of a much larger monetary value than a client's norm, and collective anomalies, where groups of consecutive data points do not follow the general pattern, e.g., a period of reduced traffic intensity due to a car accident. Anomalies frequently arise in data as symptoms of undesirable events such as malicious activity, external attacks, emergencies, system faults, etc. Thus, detecting and diagnosing anomalies in a timely manner can help ensure the safety, efficiency, and optimality of critical systems [1,3,5].

Traditionally, anomaly detection methods have been classified into three groups based on the level of external information provided: *supervised*, where examples of previous (labelled) anomalies are used as a basis for detection; *unsupervised*, where no such information is available and artefacts are identified purely on the basis of their deviation from the bulk

of the data; and *semi-supervised*, where some intermediate level of information is provided, e.g., labelling of portions of data that are definitively non-anomalous. Since supervised and semi-supervised approaches typically necessitate some human effort in segregating and labelling anomalies (which can be expensive and incompatible with large, streaming, multivariate data settings), much of modern anomaly detection, including this work, focuses on the unsupervised setting. However, most of the current unsupervised anomaly detection approaches fail on non-stationary signals, as the distribution of normal data points may shift with time but the approaches' partitioning methods are static [7–9]. Finally, it is worth mentioning in some other works [3,9,10], which explored anomaly detection in nonlinear signals via data distribution estimation or next state prediction. These methods can handle the task of separating anomalies from a changing baseline. However, they do not address the problem of classifying and identifying the different types of anomalies for early intervention.

We propose a new three-part framework to *detect and cluster* anomalies in non-stationary time series which can be used for unsupervised anomaly detection and analysis. We combine an online NARX (nonlinear autoregressive with exogenous input) detrending regime with anomaly detection (via SCAPA (sequential collective and point anomaly)) and a fuzzy clustering algorithm. NARX models are widely implemented due to their robust capabilities in the representation of complex systems that the standard ARIMA approaches cannot model [11]. SCAPA [12–14] simultaneously detects point anomalies and anomalous regions (*collective anomalies*) in the detrended series via an efficient dynamic programming regime. Finally, collective anomaly segments can be clustered via fuzzy c-means clustering [15] of their empirical distribution functions (ECDFs).

In summary, our main contribution is the provision of a method (for the first time, as far as we are aware) that considers the entire end-to-end process of

1. Detrending non-stationary time series,
2. Detecting anomalies in the residual series,
3. Grouping similar anomalies via fuzzy clustering.

Each phase uses state-of-the-art techniques from its requisite literature and can be applied effectively to multivariate series across various variants of non-stationarity and many classes of anomalous artefacts.

We provide a detailed empirical investigation of its efficacy using two datasets: the first, a year-long univariate nonlinear time series representing traffic on a telecommunications network, and the second, a multivariate real-data example from the Skoltech anomaly benchmark (SKAB) [16] pertaining to systems control.

The remaining sections are organised as follows. Section 2 outlines existing techniques and the main building blocks of our framework, while Section 3 defines our integrated approach. Our empirical results, firstly for the detection phase, are presented in Section 4, with a detailed consideration of the clustering aspect following in Sections 5–7, before the final conclusions are presented in Section 8.

2. Related Work

Anomaly detection is one of several common mining tasks for extracting meaningful knowledge from time series data [9]. However, non-stationary time series pose significant challenges for anomaly detection as it is rather more difficult to detect and interpret anomalous changes from a moving series than a stationary one, especially in real time [6–8]. Most existing methods can be categorised as identifying anomalies via model-based classification or statistical anomaly detection methods.

Model-based anomaly detection approaches can be further divided into two groups: estimation models and prediction models. Estimation models detect anomalies with the best-fit models, while prediction models focus on the prediction of the sequential states of the time series and can detect anomalies in an autoregressive way [9,17,18]. All these model-based methods [1,3–5] rely heavily on accurately labelled training data to generate clusters of 'anomalous' and 'normal' data to which subsequent data are assigned. However,

such labelled data are often expensive and hard to obtain in real time. Compared to our approach, these techniques retain little information on the structure of anomalies which would admit further analysis, such as clustering [6–8].

Statistical anomaly detection methods mainly use density-based approaches or histogram-based approaches. Density-based anomaly detection methods are based on the concept of the neighbourhood, which is more complex for ordered data. Therefore, these methods are rarely used for temporal data. Histogram-based anomaly approaches create histogram representations of the original time series and remove anomalies to achieve lower error in these representations [9,10]. Statistical anomaly detection methods work in an unsupervised manner using, for instance, distribution-based clustering [2,19,20]. However, these methods are limited by the assumption that the 'normal' behaviour is stationary or has a fixed periodic pattern which may be uncommon in practice [6–8].

Our method carries the advantages of both approaches by first detrending the data and then applying anomaly detection on the (stationary) residual series, without the need for labelled 'training' anomalies. It has the additional advantage of being able to cluster multiple anomalies based on the distribution of the residuals within. We proceed to describe the constituent parts of our approach in more detail.

2.1. Nonlinear Autoregressive with Exogenous Input Model

A NARX model allows the combination of autoregressive modelling, regression on external series, and nonlinear basis functions, and are widely-used, versatile models for the analysis of complex time series [21–24]. We deploy NARX with neural basis functions favouring their flexibility, although other options such as polynomial or wavelet bases, sigmoid networks, etc., would be equally viable [11].

The typical NARX neural network model relates a target series $\{y_t\}_{t\in\mathbb{N}}$ to an exogenous series $\{x_t\}_{t\in\mathbb{N}}$ in an autoregressive manner,

$$y_t = F^{\ell}(y_{t-1}, \ldots, y_{t-n_y}, x_t, x_{t-1}, \ldots, x_{t-n_x}) + \varepsilon_t,\ t \in \mathbb{N}, \tag{1}$$

where $n_y, n_x \in \mathbb{N}$ are the maximum lags for the series and $\varepsilon_t \in \mathbb{R}$ denotes additive noise [11]. F^{ℓ} represents the basis function—in this case, the neural network—with a nonlinearity degree of $\ell \in \mathbb{N}$, defined by the complexity of the network.

NARX neural network models have been applied to meteorological prediction tasks [25], simulation in health applications [26,27], and anomaly detection in industrial monitoring [28–30]. However, previous anomaly detection schemes have used crude threshold exceedance-based criteria which can cope poorly with drifts and shifts in variance. Our combination of NARX with SCAPA (as described in the next section) adapts more readily to non-stationary data.

2.2. Sequential Collective and Point Anomalies

SCAPA [13] is an online anomaly detection method which detects and distinguishes point and collective anomalies within time series with stationary baselines [12].

In the univariate setting, where a series $u_t \sim D(\theta(t))$ follows some distribution D, collective anomalies can be regarded as epidemic changes in parameters $\theta(t)$ away from normal parameters θ_0 [12]. SCAPA identifies collective anomalies by minimising the following penalised cost over the number of collective anomalies K, their start and end points, s_i and e_i, and normal feature parameters θ_0 [12]:

$$\min_{K, s_{1:K}, e_{1:K}, \theta_0} \sum_{t \notin \cup [s_i+1, e_i]} C(u_t, \theta_0) + \sum_{j=1}^{K} \left[\min_{\theta_j} \left(\sum_{t=s_j+1}^{e_j} C(u_t, \theta_j) \right) + \beta_c \right], \tag{2}$$

where $C(.,.)$ is a cost function, and $\beta_c > 0$ is a penalty to prevent overfitting.

In our illustrative examples, we consider the mean and variance as our feature parameters and adopt twice the negative of the Gaussian log-likelihood as our cost function—a

common choice, appropriate for our residual series. Within-anomaly parameter estimates θ_j, $j = 1, \ldots, K$ are not optimisation variables but are estimated by sample statistics within the set $\{x_{s_j+1}, \ldots, x_{e_j}\}$. SCAPA has an efficient dynamic programming implementation in the R package `anomaly` [31]. We utilise this within our empirical examples with a minimum collective anomaly length of 2, a maximum anomaly length (chosen on a per-instance basis for computational efficiency), and a separate penalty $\beta_o > 0$ for point anomalies (which are regarded as changes in variance only, with a fixed length of 1).

3. Anomaly Detection and Clustering

This section outlines our proposed end-to-end framework for online anomaly detection and clustering, integrating the previously summarised components.

3.1. NARX-SCAPA

The detection component of the NARX-SCAPA framework uses a NARX neural network to provide a one-step-ahead prediction of the target time series in real time and then deploys SCAPA to identify anomalies in the prediction errors (residual) series. We have implemented this component via `NARX` from `Neural Net Time Series App` in Matlab R2021a [32] and `SCAPA` from R package `anomaly` in CRAN [31].

Figure 1 shows the example structure of the NARX neural network used to form a prediction $\hat{y}_{t+1}$ of the next point y_{t+1} based on the n previous points and their corresponding time of day: $y_{t-n}, \ldots, y_t$ and $x_{t-n}, \ldots, x_t$. The units in the hidden layer are based on the sigmoid function and the output layer uses a linear function of all hidden unit outputs. We use 1 hidden layer with 10 neurons and softmax for the output node, which follows the default setting of conventional neural networks that can balance the complexity and accuracy of the model in most cases. The operations within the hidden layer can be expressed as:

$$Y_j^h = \sigma\left(\sum_{i=1}^{L^x} x_{t-i+1} w_{ij}^x + \sum_{i=1}^{L^y} y_{t-i+1} w_{ij}^y + b_j^h\right), \tag{3}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$, $j \in \{1, \ldots, J\}$ indexes the neurons, and L^x and L^y are the maximum lags considered in the model. w^y and w^x are the weights for the target time series and the exogenous series, respectively, and b^h represents the constant intercept. The output Y_j^hs are passed through the final layer to form a prediction:

$$\hat{y}_{t+1} = \sum_{j=1}^{J} Y_j^h w_j + b_j^o. \tag{4}$$

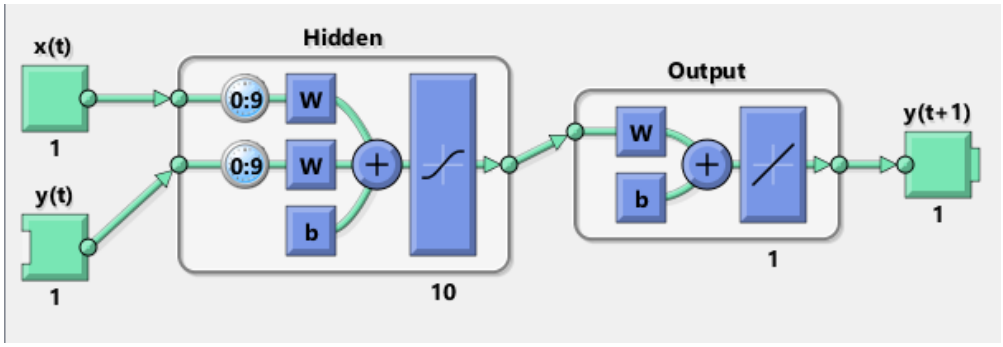


Figure 1. Structure of NARX neural network model.

Once y_{t+1} is observed, a residual $r_{t+1} = \hat{y}_{t+1} - y_{t+1}$ is computed and reported as the next real-time observation to SCAPA.

Weights are estimated from an initial period of (assumed non-anomalous and complete) training data. We replace any missing observations and SCAPA-identified anomalous

data from the (continuously evolving) input set with the prediction of the trained model. Figure 2 summarises the structure of the detection component.

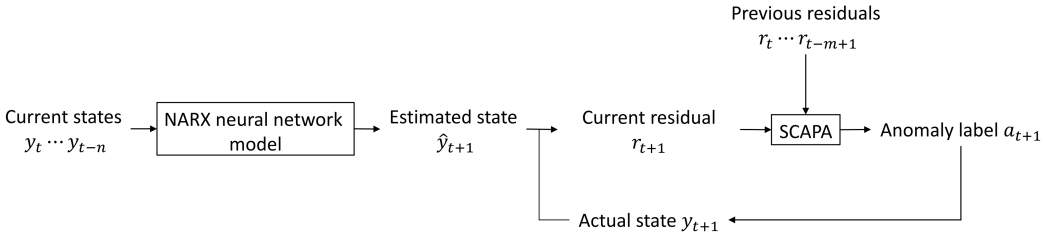


Figure 2. Structure of implemented NARX-SCAPA framework.

The following steps summarise the overall detection mechanism:

1. Initialise a NARX neural network model with the training data;
2. Calculate the estimated value for the current time based on the actual observation values of the previous 10 s;
3. Calculate the current residual value between the estimated value and the actual observation when it is obtained;
4. Pass this residual as the next streaming observation to SCAPA to detect anomalies;
5. Label any points in the observation time series as anomalous if the corresponding residuals are identified by SCAPA;
6. Repeat steps 2 to 5 until the observation stops.

3.2. ECDF-Based Fuzzy c-Means Clustering

Once multiple anomalies are detected, we may follow up with clustering. We propose a method that represents anomalous segments (of potentially different lengths) by empirical cumulative distribution functions (ECDFs). Our ECDFs are discretized bins since fuzzy c-means clustering works well with vectors. The number of bins is of course adaptable, and for the subsequent experiments, 20 bins was chosen after exploratory analysis as a value which balanced the ability to capture the structure of anomalies with the ability to make reliable inference. The result is a ECDF vector A_i associated with each anomaly $\mathbf{a}_i$, where $A_{i,j} = |\mathbf{a}_i|^{-1} \sum_{a \in \mathbf{a}_i} \mathbb{I}\{a \leq -1 + 0.1j\}$ for $j = 1, \ldots, 20$. Since we standardise the residuals to $[-1, 1]$, we always have $A_{i,20} = 1$.

Fuzzy c-means clustering is a soft partitioning method whose output assigns each vector A_i, $i = 1, \ldots, N$ a membership degree $\mu_{ij} \geq 0$ for each cluster $j = 1, \ldots, M$ such that $\sum_{j=1}^{M} \mu_{ij} = 1$ [15].

The clustering is chosen as a solution to the optimisation problem:

$$\min_{\mu} \mathcal{J}_m = \sum_{i=1}^{D} \sum_{j=1}^{N} \mu_{ij}^m ||A_i - C_j||^2, \tag{5}$$

where D and N are the numbers of ECDF vectors and clusters, respectively. The parameter m is a fuzzy partition exponent for the degree of fuzzy cluster overlap and is set to be 2, based on the default setting of common fuzzy clustering. The A_i and C_j represent the ith ECDF vector for anomalies and the jth cluster centre vector.

The following steps explain the fuzzy clustering mechanism [15]:

1. Randomly initialise the degree of membership μ_{ij}.
2. Calculate the cluster centre vectors with:

$$Cj = \frac{\sum_{i=1}^{D} \mu_{ij} A_i}{\sum_{i=1}^{D} \mu_{ij}}, \quad j = 1, \ldots, N. \tag{6}$$

3. Update the degree of membership using:

$$\mu_{ij} = \left(\sum_{l=1}^{N} \left(\frac{||A_i - C_j||}{||A_i - C_k||} \right)^2 \right)^{-1} \quad i = 1, \ldots, D, \; j = 1, \ldots, N. \tag{7}$$

4. Compute $\mathcal{J}_m$ with updated centres and membership degrees.
5. Repeat steps 2 to 4 until the distance between the old and new objective function is smaller than the minimum improvement threshold.

4. Detection Results

In this section, we illustrate the effectiveness of our approach on a telecommunications traffic dataset developed with BT.

4.1. Time Series Data

The BT network traffic data is a long-term non-stationary time series with a repeated daily pattern. It records traffic from BT's network for continuous days at a sampling rate of 1 Hz. Within the data, some several anomalous points and regions do not follow the general distribution of the rest of the series, which in practice may only be partially identified and labelled by human experts and engineers or difficult to identify reliably at scale. Figure 3 shows an excerpt of the time series from Day 1 to Day 5. The traffic follows a similar daily pattern due to the regular fluctuation of general uses but can include anomalies and noise on the daily level—as well as longer scale drifts and changes.

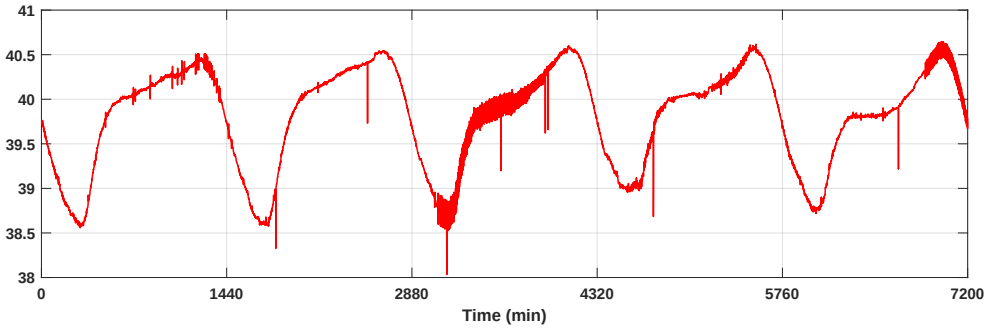


Figure 3. Original BT network traffic data from Day 1 to Day 5.

Drifts and shifts away from the established pattern as well as severe periods of interference are considered anomalies and can be consistently correlated with serious operational issues by engineers. As such, they often necessitate intervention, which may vary depending on the nature of the anomaly. Therefore, being able to detect and identify different anomalies in an online fashion plays a crucial role in maintaining the normal operational state of the network.

For confidentiality and to allow verification of our results, the bulk of our experimental work is based on a series simulated to have similar properties. We created a smooth series of 100 days of data to which we added artificial anomalies in known locations. The simulated series carries a similar daily fluctuation. The artificially introduced anomalies are also inspired by the structure of those manually labelled in the original series. Point anomalies are added by shifting random points up or down by a value sampled uniformly from $[0.5, 1]$. Our collective anomalies can be separated into three main types: (1) Splits: the trajectory of the time series splits into two or three parallel lines with spacing $\pm$ 0.2 to 0.3 for 100 to 200 min. (2) Interference: the variance of the time series significantly increases (5 to 10 times) for 75 to 250 min. (3) Shifts: the trajectory of the time series jumps to a different level and causes discontinuity, with 0.1 to 0.2 in amplitude and 0 to 10 times the original variance for 75 to 250 min.

The original series also exhibited anomalies that gradually returned to normal behaviour, as the intensity of the shift or interference slowly decrease (or decayed). These are particularly difficult to accurately detect. Based on these findings, we included 20 types of anomalies with intensities scaled between 0 and 1 in the simulation (see Table A1 in

Appendix B for details). For each anomaly type, there are 10 instances randomly distributed over the entire simulated time series.

4.2. Online Anomaly Detection

The number of neurons in the hidden layer of NARX neural network models was set to be 10, and the maximum lags were also taken to be 10. With the BT network traffic data, the time of day in minutes (i.e., 1 to 1440) was introduced as the exogenous series (with a time delay $d = 0$). The NARX model was fitted to five days of training data from the original series by optimising the weights w and constants b with Levenberg–Marquardt backpropagation [33].

For SCAPA, we set the minimum and maximum segment lengths of collective anomalies to be $l = 30$ and $m = 250$, respectively, for the BT network traffic data, and the penalties for collective and point anomalies to be $\beta_c = 75$ and $\beta_o = 25$, respectively. These choices ensured that all known anomalies in the training data perfectly matched the sets of detected anomalous points, including the long (and decaying) collective anomalies.

Figure 4 shows the results of implementing NARX-SCAPA on the first 10 days of the simulated traffic series with these settings. We see the NARX component's success in capturing the typical non-stationary pattern and generating residual series where anomalies can be detected by SCAPA. It can be observed that all the anomalies were correctly identified and labelled, even for the point anomaly around 7500 min and the collective anomaly around 8800 to 9200, which were hard to identify by eye alone but are easily observed in the residual series.

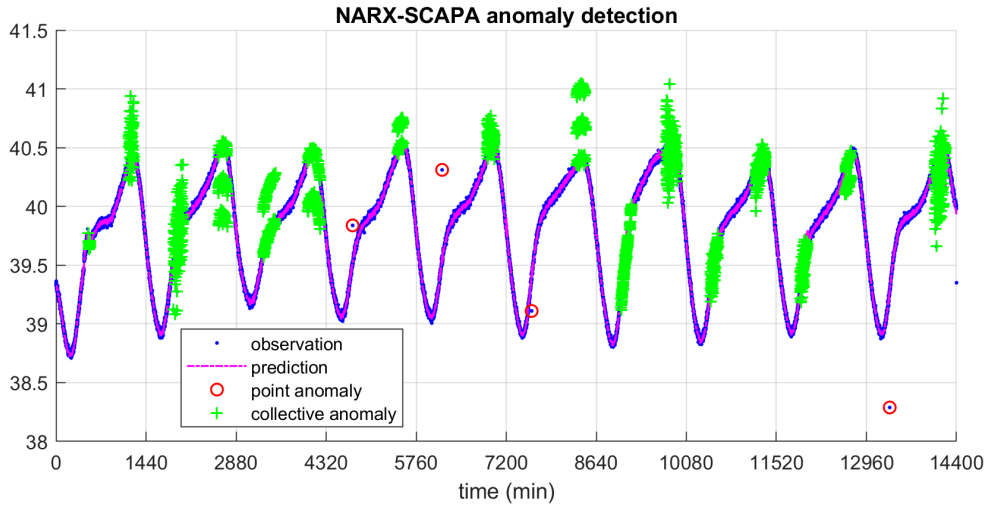


Figure 4. NARX-SCAPA anomaly detection on the simulated BT network traffic data from Day 1 to Day 10.

In the full 100 days of simulated data, we introduced a total of 24,750 data points across 160 collective anomalies NARX-SCAPA identifies a total of 22,971 points within collective anomalies, 20,569 of which match with the design. The success of detection was significantly influenced by the intensity of an anomaly and the shape of the nonlinear series at the location of the anomaly, e.g., collective anomalies which commenced at a turning point of the series were hard to detect since the contrast to the regular series was minimal.

Meanwhile, since the intensity of anomaly points declined in the tails of decaying collective anomalies, the data points towards the end of a decay anomaly were likely to be neglected by the algorithm for lack of signal strength. Also, for some regions with relatively low-intensity collective anomalies, the NARX neural network was able to predict them as if they were normal parts of the time series. As a consequence, the residual series failed to provide any structural information about correlated collective anomalies in these regions and the SCAPA failed to label them as anomalies properly.

Table 1 summarises the collective anomaly detection results of the NARX-SCAPA. Anomaly regions with low or declining intensity, such as Type 13 to Type 18, were more

difficult for the model to detect. However, the overall performance of the NARX-SCAPA model shows efficacy and efficiency in detecting and identifying various anomalies within non-stationary time series in real time. Similarly, there were a total of 40 point anomalies introduced into the simulated time series, of which NARX-SCAPA detected 33, with no false positives. Again, the success of detection depended on the intensity of the anomaly, as the breakdown by type shows.

Table 1. NARX-SCAPA anomaly detection on the simulated BT network traffic data (see Table A1 in Appendix B for details of anomaly types).

				Split				Interference		
Type	1	2	3	4	5	6	7	8	9	10
Actual points	1000	1500	2000	1000	1500	2000	750	1500	2000	2500
Identified points	974	1490	1953	982	1499	1856	743	1491	1354	2089
Accuracy (%)	97.4	99.3	97.7	98.2	99.9	92.8	99.1	99.4	67.7	83.6
				Shift				Point		
Type	11	12	13	14	15	16	17	18	19	20
Actual points	750	1500	2000	2500	750	1500	10	10	10	10
Identified Points	320	778	1128	1666	748	1498	6	10	7	10
Accuracy (%)	42.7	51.9	68.5	66.6	99.7	99.9	60.0	100	70.0	100

4.3. Real System Implementation

We also illustrate the performance of NARX-SCAPA on the original (i.e., not simulated) time series. Again, the model shows success at one-step ahead prediction in non-anomalous regions, and anomaly detection based on residuals. The NARX neural network was capable of handling the non-stationary time series and was adaptive to fluctuations within the normal range. Moreover, the model remained robust to the impact of severe anomalies, e.g., the regions around 7300, 11,000, and 13,000 min, where the observations were scattered, interfered, and split, respectively. The structures and the intensities of these anomalous regions were retained in the detrended residual series. However, it can also be seen in Figure 5 that the prediction of the NARX-SCAPA-based model was distorted between 13,250 to 13,400 min, and this was due to the limitation of the NARX model and the long duration of strong anomalies. Yet, such distortions of the model prediction in the anomaly region were not able to compromise the detection of anomalies, as seen in Figure 5. We see success in detecting point and collective anomalies, including those low-intensity anomalies which are hard to observe by eye alone.

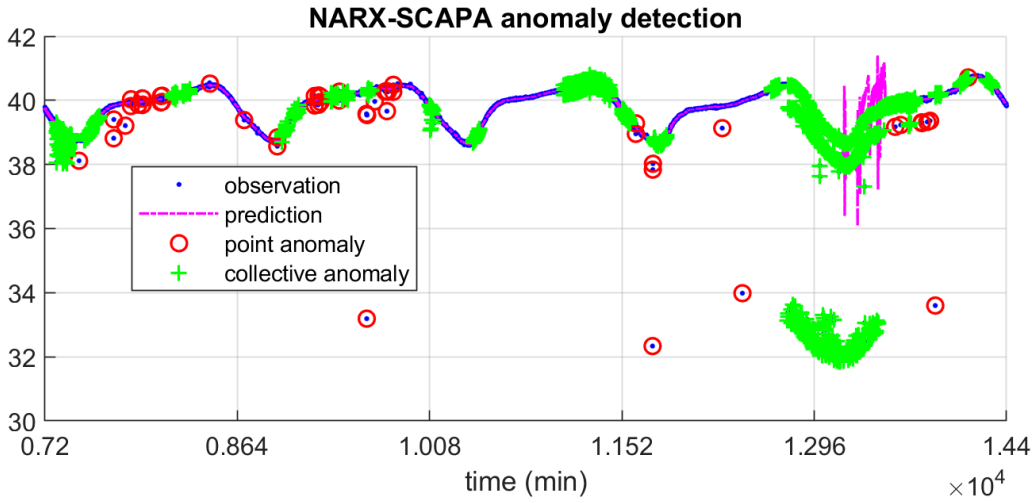


Figure 5. NARX-SCAPA anomaly detection on the original BT network traffic data.

4.4. Alternative Approaches

We compare the performance of NARX-SCAPA with two popular, existing alternatives from the R packages `Forecast` and `HDoutlier`. The `Forecast` package, based on the model estimation method, is a commonly used tool for analysing large numbers of univariate time series in real time [34]. It uses the `tsoutliers` method to model univariate non-stationary time series via Multiple Seasonal-Trend decomposition using Loess (MSTL) and Friedman's super smoother [35]. Possible anomalies are identified within the two most extreme quartiles of the residual series. The `HDoutlier` method allows probabilistic labelling of anomalies [36]. The method partitions data into different exemplars according to the length of the series. It then computes nearest-neighbour distances between exemplars and fits the cumulative distribution function and labels data points in any exemplar that is far from the others as anomalies.

Tables 2 and 3 numerically compare the performance of these alternatives to NARX-SCAPA on the simulated data, both applied directly to the raw data and to the detrended residuals. We see that `Forecast` was more suitable for the residual series, and `HDoutlier` performed similarly in both cases. However, neither can achieve the same accuracy as NARX-SCAPA for detecting collective anomalies. `HDoutlier` achieved high accuracy for point anomalies, but this was coupled with an extremely high false positive rate.

Table 2. Accuracy of `Forecast` and `HDOutlier` methods on simulated BT traffic time series. We report performance on raw observations (Obs) and NARX residuals (Res). All accuracies are percentages.

Anomaly Type	1	2	3	4	5	6	7	8	9	10
`Forecast`—Obs	29.0	1.9	10.9	3.2	24.9	1.2	3.0	16.6	8.0	9.8
`Forecast`—Res	50.7	75.3	48.9	53.1	76.7	50.5	29.7	57.2	8.7	32.2
`HDOutlier`—Obs	67.0	76.8	66.8	68.3	75.2	67.3	81.9	91.7	69.1	79.6
`HDOutlier`—Res	64.9	75.0	65.0	66.7	75.3	67.0	82.7	90.9	68.0	78.7
Anomaly Type	11	12	13	14	15	16	17	18	19	20
`Forecast`—Obs	41.1	0.0	14.1	2.6	27.6	5.5	30.0	60.0	0.0	0.0
`Forecast`—Res	5.5	7.1	13.9	14.6	66.3	63.5	70.0	100	80.0	100
`HDOutlier`—Obs	58.7	58.3	65.6	69.1	92.3	90.4	100	100	90.0	100
`HDOutlier`—Res	60.7	62.3	68.9	68.1	91.3	86.7	100	100	90.0	100

Table 3. Performance summary of anomaly detection with NARX-SCAPA, `Forecast` and `HDoutlier` on the simulated BT traffic time series.

Performance Rate	**NARX-SCAPA**	**Forecast**		**HDoutlier**	
		Observation	**Residual**	**Observation**	**Residual**
True negative	98.0%	98.5%	99.6%	54.8%	54.9%
True positive	83.1%	10.3%	38.9%	72.8%	73.2%
False negative	16.9%	89.7%	61.1%	27.2%	26.8%
False positive	2.0%	1.5%	0.4%	45.2%	45.1%

Figures A1 and A2 in Appendix A provide a fuller visualisation and discussion of the results of implementing `Forecast` and `HDoutlier` approaches on the original BT network traffic data. We see that the `tsoutliers` algorithm was overly sensitive in handling the residuals, and `HDoutliers` was more conservative. It is worth noting that, for the BT traffic data, the probability of anomalies is low, yet their intensity is high. Therefore, `Forecast` does not work well with a fixed strength of seasonality for the decomposition. Meanwhile, collective anomalies of the split Type remain a challenge for `HDoutliers` as it either treats the outer data points as anomalies or labels entire regions as anomalies with a high false positive rate.

Overall, compared to the performance of `Forecast` and `HDoutlier`, the proposed NARX-SCAPA model has achieved the best anomaly detection results with minimal false positives and false negatives (see Table 3).

5. Offline Anomaly Clustering

In this section, we consider the problem of clustering the detected collective anomalies based on the residuals. Clustering collective anomalies of different lengths based on their intensity and structure is a non-trivial task in itself, without an online approach. Therefore, we first normalise the original residual sequences with different distributional summaries (histogram and ECDF). Then, since the clustering usually depends on the distances between the targets, we explore two distance measures (Jensen–Shannon divergence and Euclidean distance), and finally, we compare various partitioning schemes (hierarchical clustering, k-means clustering, fuzzy c-means clustering, EPmeans and HMCluster).

5.1. Known Cluster Numbers

We deploy these combinations to cluster anomalies detected from running NARX-SCAPA on the simulated data, with the constraint that they should find 16 clusters. We compare performance across methods via the Rand index [37] which counts the proportion of points allocated to the 'correct' cluster for their anomaly type. While the true anomaly type for each anomalous point is known to us, the clustering algorithms approach the problem as an unsupervised one and there are false positives among the anomalous points. Thus, to compute a performance measure, such as the Rand index, we must associate each cluster obtained with an anomaly type 1–16. This is achieved via a linear assignment algorithm [38] which chooses an anomaly type to associate with each cluster, based on the proportions of each anomaly type therein.

The clustering algorithms take vectorised distribution summaries as their input, rather than the (variably sized) sets of residuals within each collective anomaly. We consider a histogram (or bin-wise frequency) summary and a discretised empirical cumulative distribution function (ECDF). Within the clustering algorithms, we compare two distance measures: the Jensen–Shannon divergence and the Euclidean distance. Under the hierarchical clustering, the histogram method achieves a Rand index of 0.51 with the Jensen–Shannon divergence and 0.43 with the Euclidean distance, and the ECDF method has a Rand index of 0.49 with Jensen–Shannon divergence and 0.44 with Euclidean distance.

We therefore argue that with hierarchical clustering, compared to Euclidean distance, Jensen–Shannon divergence is a more suitable measurement for the similarity of two probability distributions. We did however find that it was more sensitive to the granularity of the discretisation, with overly coarse choices leading to unreliable results, while the algorithm using Euclidean distance remained more robust. Furthermore, due to the complexity of differentiation within the expectation-maximisation iteration, Jensen–Shannon divergence cannot be directly combined with partitioning-based clustering algorithms, e.g., k-means clustering and fuzzy c-means clustering. Meanwhile, we see in Table 4 that the ECDF works better than histogram with various clustering methods in distinguishing different types of collective anomalies in time series.

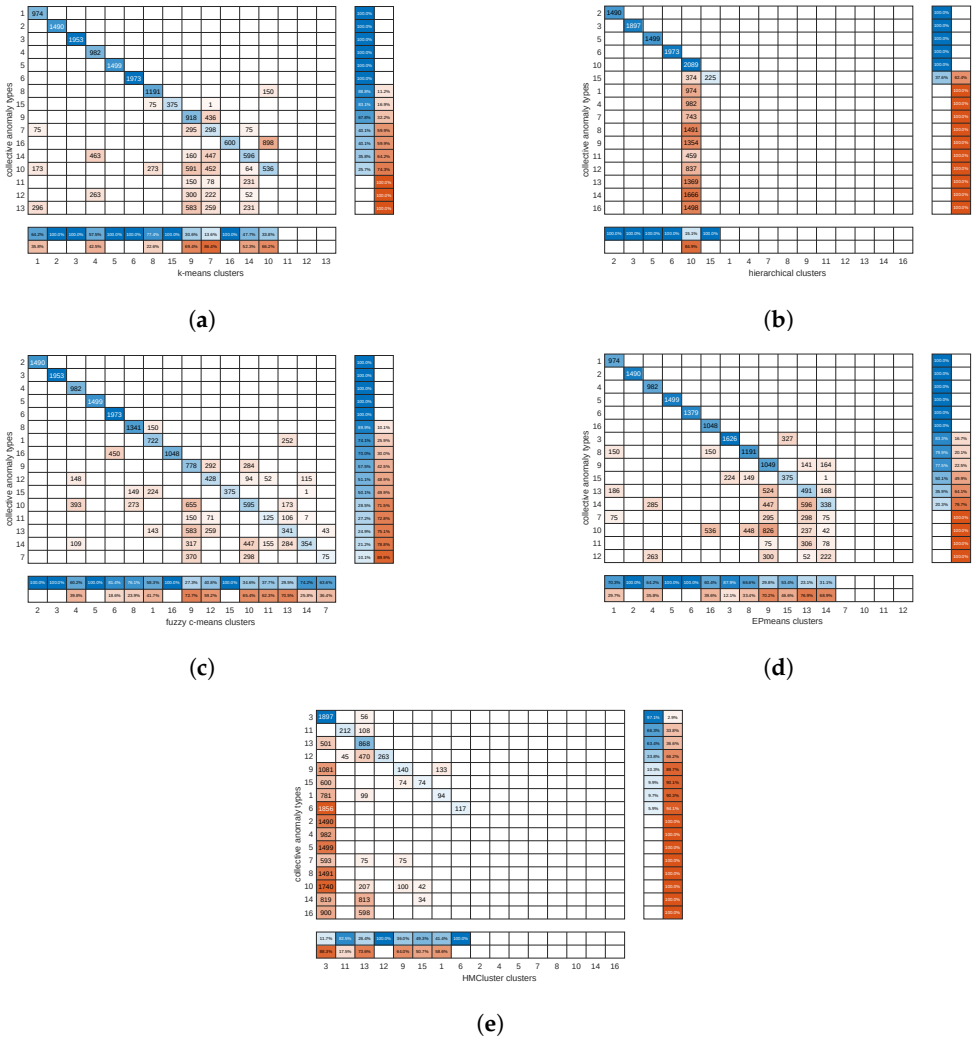
Table 4. Rand index comparison of distribution summaries and clustering methods.

	Hierarchical	**K-Means**	**Fuzzy c-Means**	**EPmeans**	**HMCluster**
Histogram	0.43	0.59	0.56		
ECDF	0.44	0.62	0.68	0.62	
Spectral					0.18

Next, we have compared the performances of three commonly used clustering algorithms with Euclidean distance: hierarchical clustering, k-means clustering and fuzzy c-means clustering. K-means clustering is a well-known spatial clustering algorithm that bases on the expectation-maximisation iteration to compute k clusters with all data points assigned to their nearest cluster centroids [39]. Hierarchical clustering builds on the hierarchy of clusters that either merge or split data points with dendrograms [40]. In addition, we have also applied two state-of-the-art clustering methods that were specifically designed for time series clustering: EPmeans and HMCluster. EPmeans is a non-parametric clustering approach

that combines the Earth Mover's Distance and k-means clustering for clustering probability distribution based on empirical cumulative distribution function (ECDF) [41]. HMCluster is a spectral theory-based hierarchical method for identifying time series with total variation distance [42]. The clustering performances using different distribution summaries and clustering methods are summarised with the Rand index in Table 4.

The partitioning-based clustering methods—k-means clustering, fuzzy c-means clustering and EPmeans—have outperformed the tree-based clustering methods—hierarchical clustering and HMCluster. It is worth noting that hierarchical clustering is sensitive to noise and outliers. Meanwhile, for handling clusters with different sizes, hierarchical clustering tends to split large clusters and merge small clusters. Meanwhile, the results in Table 4 show that, compared to the histogram, an ECDF can directly reflect the structures of time series segments without binning bias and be more robust to the outliers. Therefore, the ECDF appears to be a more suitable distribution summary for clustering collective anomalies. Compared to the other methods, the fuzzy c-means clustering with ECDF has achieved the best results for identifying all 16 types of anomalies with the least error rates. We summarise the clustering results using ECDF with confusion matrices in Figure 6a–e, which allows for a more detailed comparison of different clustering methods.



(**a**) (**b**)

(**c**) (**d**)

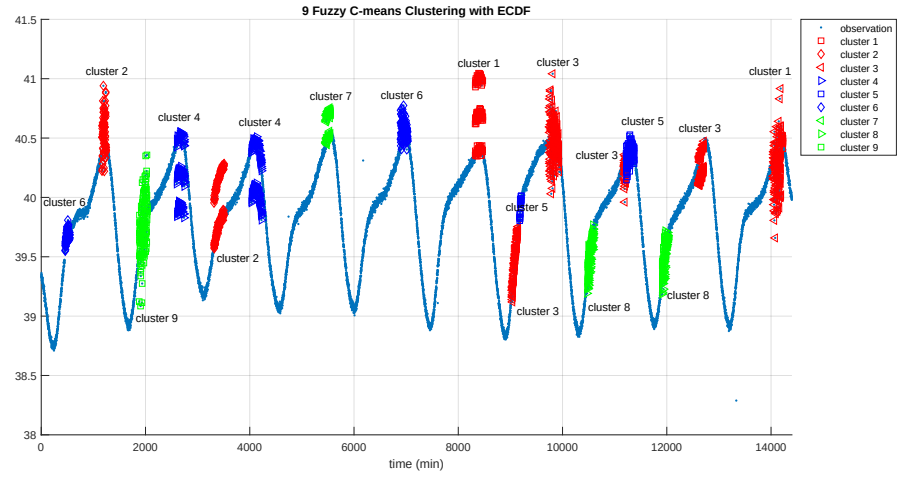
(**e**)

Figure 6. Clustering results by method presented as contingency matrices over 16 collective anomaly types. (**a**) K-means clustering with ECDF for 16 types of collective anomalies. (**b**) Hierarchical clustering with ECDF for 16 types of collective anomalies. (**c**) Fuzzy c-means clustering with ECDF for 16 types of collective anomalies. (**d**) EPmeans for 16 types of collective anomalies. (**e**) HMCluster for 16 types of collective anomalies.

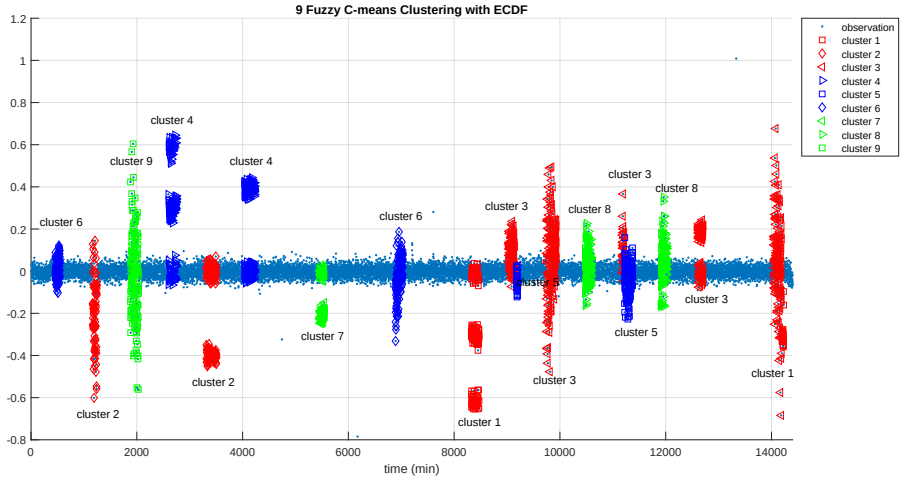
Fuzzy c-means clustering with ECDF has outperformed other clustering methods, especially for differentiating shifts with small intensity from the interference (seeing Figure 6c). EPmeans has achieved the second-best accuracy, only having some issues mixing the interference and the splits as in Figure 6d. As shown in Figure 6a, conventional k-means clustering shows reasonable clustering results, successfully dividing segments with different structures but still failing to differentiate the anomalies with varying intensity. Hierarchical-based clustering methods give the worst outcome as they could neither differentiate based on structure nor intensity of anomalies, as seen in Figure 6b,e.

5.2. Sensitivity to Algorithm Parameters

To assess the sensitivity of the 'optimal' clustering methodology—fuzzy c-means clustering with ECDF using Euclidean distance—we conduct further tests on the simulated time series with an assumption of nine clusters. Figure 7 illustrates the results on 10 days of the series. We see that the three main categories of collective anomalies—split, interference, and shift—are appropriately distinguished. This suggests that the combination of fuzzy c-means clustering and ECDF is an effective way to cluster anomalous time series segments.



(**a**)



(**b**)

Figure 7. Fuzzy c-means clustering with ECDF for 9 clusters. (**a**) Time series. (**b**) Residual series.

The maximum anomaly length of SCAPA is another key tunable parameter, so we experiment with it reduced to 60. This is less than the true maximum length and thus collective anomalies in the simulated time series may be sliced into multiple segments or shortened by SCAPA missing out the tails. Figure 8 illustrates the results with the

reduced maximum anomaly length and its resultant fragmented anomalies. We see a more pronounced detrimental effect as a result of the fragmentation. Principally, this is because tails of decaying anomalies were ignored or assigned to different anomaly segments by SCAPA, and points from the same anomaly type can seem to belong to very different patterns. Nevertheless, fuzzy c-means clustering with ECDF is able to cluster the most segments with similar structures together correctly, except for the region around 3250 min, where there is a change of nonlinear pattern for the time series.

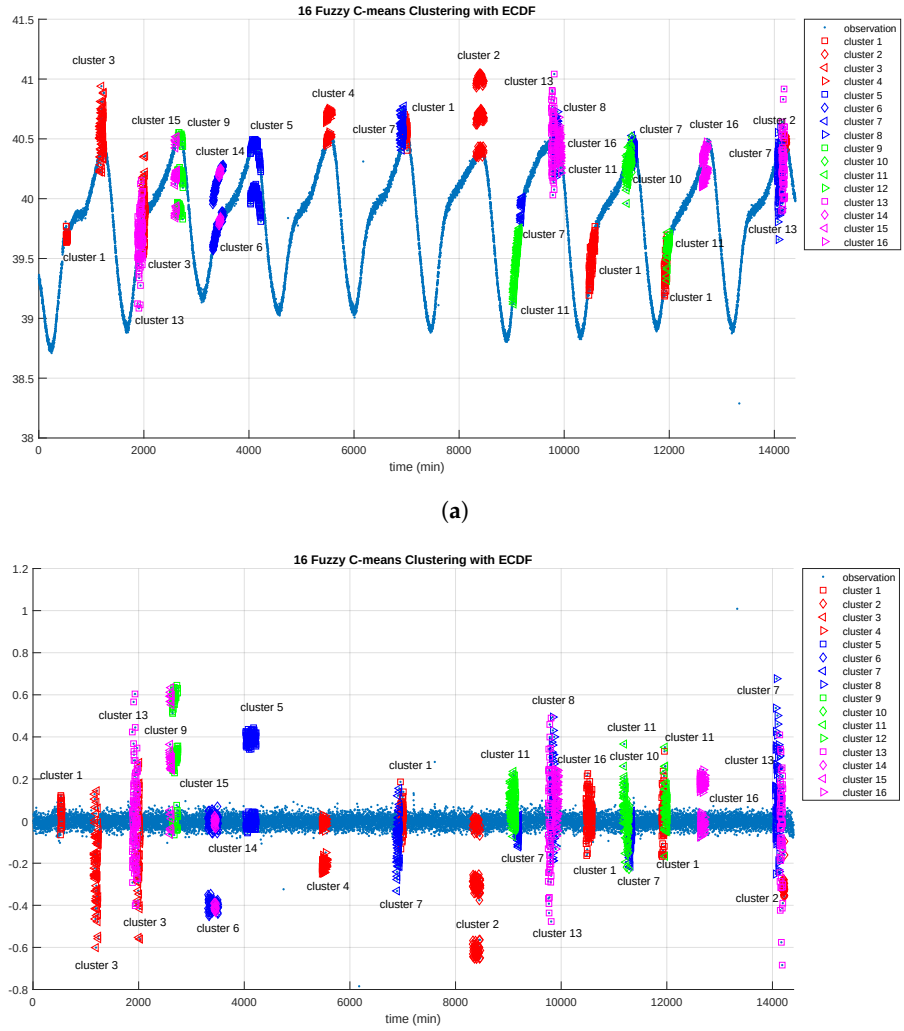


Figure 8. Fuzzy c-means clustering with ECDF on fragmented segments. (**a**) Time series. (**b**) Residual series.

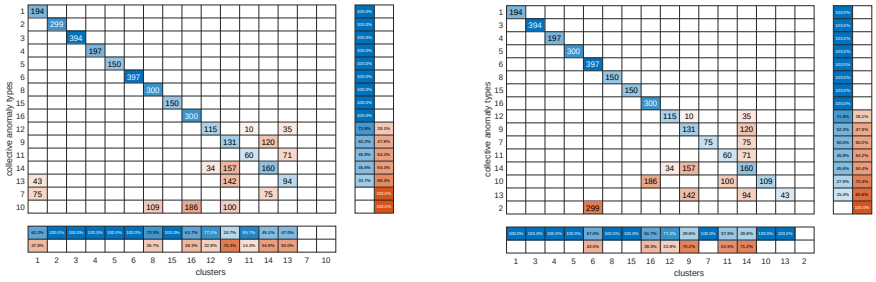
Generally, we see that fuzzy c-means clustering and ECDF is an effective clustering method that can perform well with fragmented anomalous time series segments. It is more sensitive to the structural information rather than the general intensity of data distributions. Consequently, its performance was robust to some misspecification of the number of clusters but is slightly compromised with the misspecification of segment length.

6. Online Anomaly Clustering

In this section, we propose an integration of NARX-SCAPA and the ECDF-based fuzzy c-means clustering, so that anomalies can be clustered online. Such access to real-time clustering of anomalies can help decision-makers to diagnose and act on a system more effectively by making associations with previous anomalies.

We test the real-time clustering of collective anomalies with clusters from fuzzy c-means clustering using ECDF in a simple clustering/allocation scheme. First, we evenly divide the 20-day time series data into a training group and a testing group. Next, we detect the collective anomalies within the entire training group using the NARX-SCPA framework, and based on the results, we generate anomaly clusters with the ECDF-based fuzzy c-means clustering offline. Then, we implement the same anomaly detection method on the testing group in real time, and when a new anomaly sequence is fully detected, it is assigned to the nearest cluster in terms of the Euclidean distance.

The results of such a clustering/allocation scheme are compared to the clustering results of fuzzy c-means clustering using ECDF with all anomalous time series segments from the 20-day simulated time series. It can be found in Figure 9 that the performance of the clustering/allocation approach is similar to the clustering performance of fuzzy c-means clustering with ECDF, with Rand indices of 0.69 versus 0.65. It suggests that such a clustering/allocation scheme is capable of online anomaly clustering with fuzzy c-means clustering, Euclidean distance, and ECDF. However, it is also worth noting that an increase in the amount of training data in the clustering stage can help the clustering with more generalised high-quality clusters that can accommodate more possible individual differences between the anomalies of the same type.



(**a**) (**b**)

Figure 9. Fuzzy c-means clustering with ECDF. (**a**) Online clustering with 10 days of data for clustering and the following 10 days of data for allocation. (**b**) Offline clustering with 20 days data.

7. Multichannel Data

The Skoltech anomaly benchmark (SKAB) contains multichannel time series for evaluation of anomaly detection algorithms [16]. It consists of multiple instances of multivariate series associated with eight sensors monitoring conditions on an IIot testbed and two additional series indicating the presence of anomalies and change points, as illustrated in the example in Figure 10.

To model each channel of SKAB, we use the time series of other channels as the exogenous series to predict the target time series (also with the time delay $d = 0$). Again, the model is optimised with Levenberg–Marquardt backpropagation. The anomaly-free instances in SKAB are used as training data for the modelling. Meanwhile, the minimum and maximum segment lengths of collective anomalies are $l = 100$ and $m = 500$, respectively, and the penalties for collective and point anomalies to be $\beta_c = 75$ and $\beta_o = 2.5$, respectively. The parameter settings are tuned to ensure no labelling of non-anomalous data as anomalous. To cluster, we use fuzzy c-means clustering with ECDF and optimise for three clusters. There are a total number of 7826 data points within collective anomalies across a total of 22,274 data points in SKAB. NARX-SCAPA identifies 5698 of these, achieving an accuracy of 72.8%. Figure 11 shows the results of anomaly detection and clustering on an excerpt of the SKAB data. According to the anomaly indicators in the bottom-right plot, all anomaly regions are correctly detected and clustered according to the severity (an increase from green to yellow to red). It is worth noting that the original datasets do not provide any additional information about the anomaly types. Therefore, our clustering is mainly based on the structures of detected anomalies, and the severity of anomalies depends on the distance between the detected sequence and the baseline.

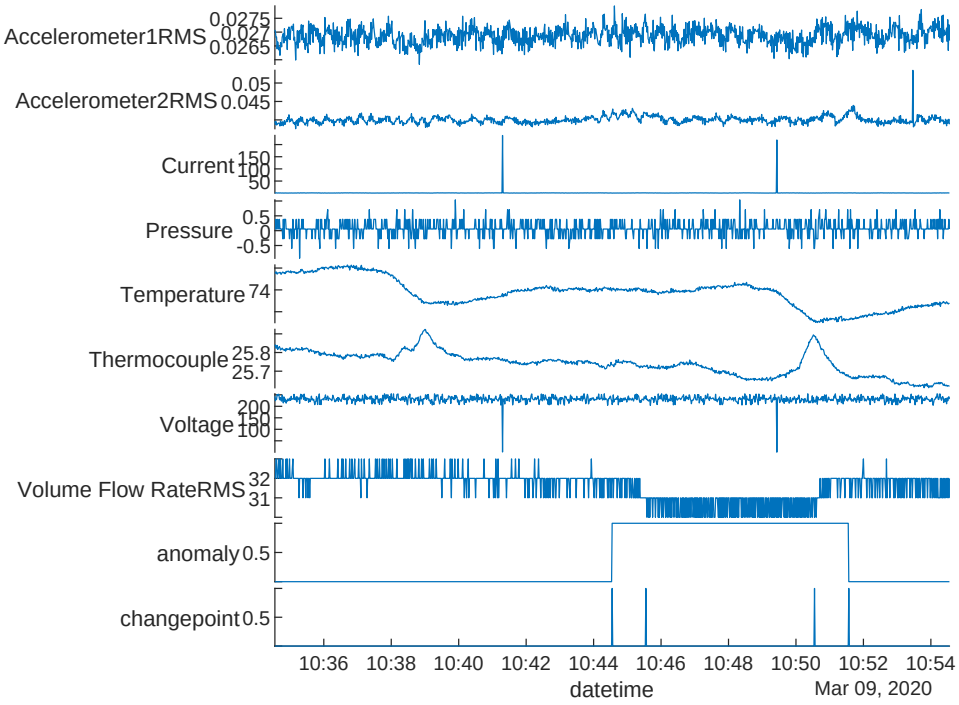


Figure 10. Example of the multichannel time series from SKAB.

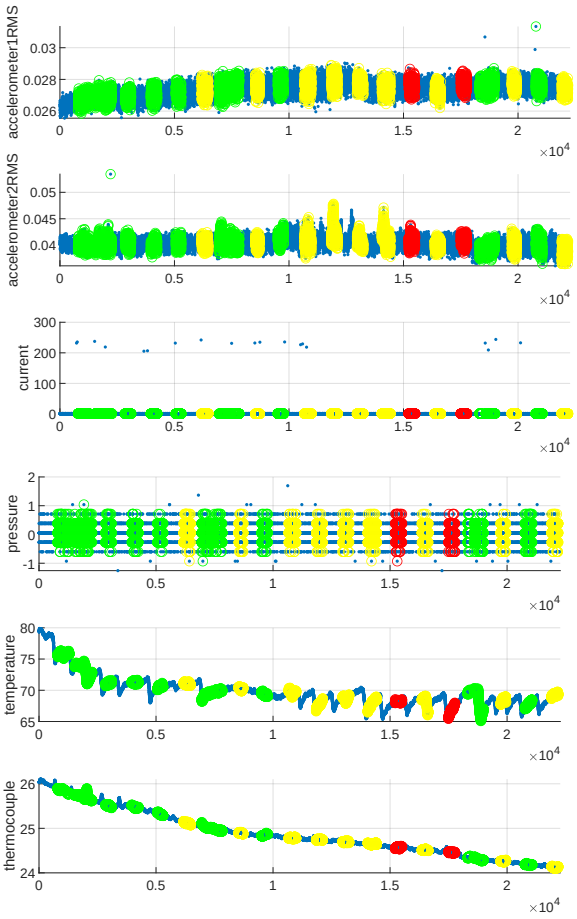


Figure 11. *Cont*.

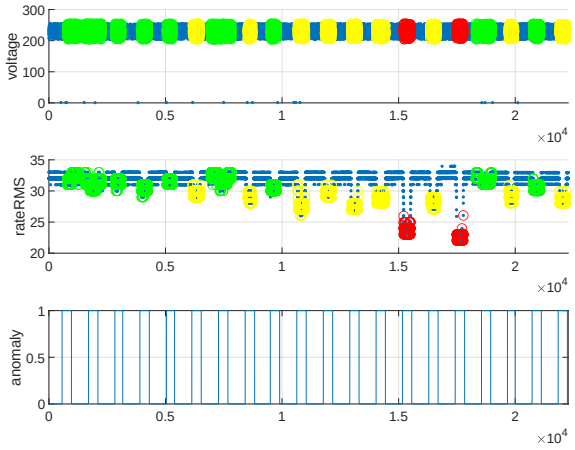


Figure 11. Anomaly detection and clustering for SKAB.

8. Conclusions

This work integrates the NARX model with SCAPA, and such a combination has outperformed the conventional methods for the detection and identification of point and collective anomalies within real-time non-stationary time series. The autoregression approach of the model is capable of handling variance forms of anomalies within an online setting, and the statistical anomaly detection provides the change of mean and variance for each anomalous point. Furthermore, it explores the clustering of collective anomalies within the time series using an ECDF-based fuzzy c-means clustering approach. Such a paradigm can create a foundation for more advanced anomaly analysis and decision-making for online systems. The proposed methods can also be applied to many diverse areas such as sensor monitoring, traffic control, and network optimisation.

Author Contributions: Conceptualization, data curation, funding acquisition, and project administration, D.S.L.; investigation, methodology, software, visualization, and writing—original draft, C.H.; resources, supervision, and writing—review and editing, D.S.L. and J.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Next-Generation Converged Digital Infrastructure project EP/R004935/1.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data used in this study includes private data from the UK's national communications network and public data from Kaggle, which the latter can be found at https://www.kaggle.com/datasets/yuriykatser/skoltech-anomaly-benchmark-skab (accessed on 16 January 2023). The collection and use of the data follow the regulation of BT Group plc.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the analyses, or interpretation of data; or in the writing of the manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

ECDF	Empirical Cumulative Distribution Function
NARX	Nonlinear Autoregressive with Exogeneous Input
SCAPA	Sequential Collective and Point Anomaly
SKAB	Skoltech Anomaly Benchmark

Appendix A. Alternative Anomaly Detection Methods

Figure A1 shows the anomaly detection results of implementing the `Forecast` model on the original BT network traffic data. Since most anomaly detection methods work better with stationary data, we have applied this method on both the original BT non-stationary

traffic time series and the detrended residual series from the NARX neural network for a better comparison. The first plot is the anomaly detection results based on the original BT network traffic data and the second plot summarises the anomaly detection results based on deflated residual time series.

The Forecast model is a widely used model estimation method for anomaly detection in many applications due to its simplicity and interpretability. It can be found that compared to the results of the NARX-SCAPA, the `Forecast` algorithm was too conservative with the original observations (e.g., anomalies with low intensity) and too sensitive with the residuals (e.g., the normal region around 12,240 min). It is worth noting that the method is based on decomposing the target time series and dividing normal and anomaly groups for the remaining time series. The detection results of this method may vary with different decomposing ratios. However, in this case, such tuning options did not show a significant impact on the detection performance. The detection only focuses on the outliers outside the baseline or the curve so the data in the split anomaly regions which are aligned with the normal pattern cannot be identified with any decomposing ratio we choose. Meanwhile, varying the decomposing ratio to its extremes can make the detection oversensitive. As is shown in Figure A1, a decomposing ratio that ensures correct labelling of split anomaly can also result in a high false positive rate. This is due to the fact that the anomalies are not evenly distributed over the series and a fixed threshold cannot adapt to it.

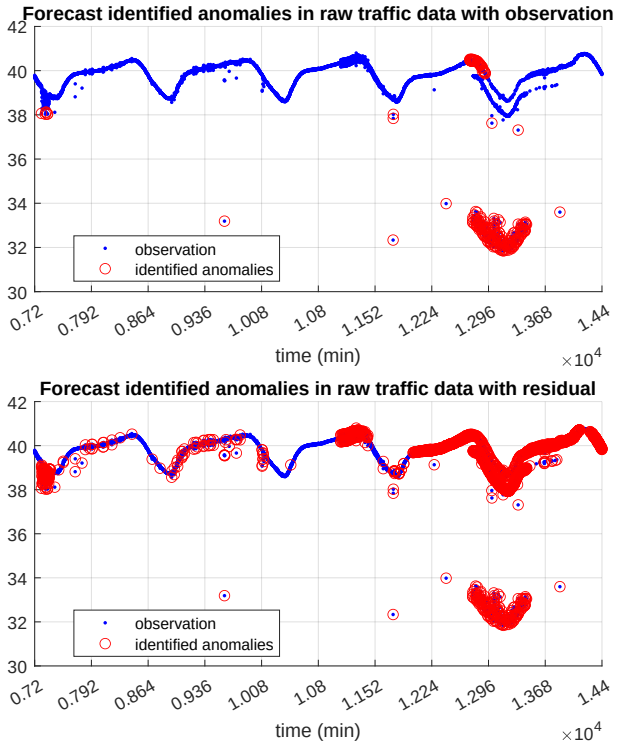


Figure A1. `Forecast` anomaly detection on the original BT network traffic data from Day 6 to Day 10.

Figure A2 shows the anomaly detection results of `HDoutlier` package for the original BT network traffic data. Similarly, the first plot represents the detection results based on the original observations and the second plot is based on the residual series from the NARX neural network. This anomaly detection method comes with a threshold α for determining the cutoff for anomalies. It can be found in Figure 5 that there is a major anomalous segment around 12,960 min. Such a segment was considered to tune the α values for `HDoutlier` on both the original time series and residual time series. In order to identify the most anomalous data within that segment, the α value for `HDoutlier` based on the original BT network traffic time series needed to be 1 (the maximum setting), and with such a setting the algorithm became over sensitive (e.g., labelling the normal region between 10,100

and 10,800 as anomalous region). However, the head and tail of the target anomalous segment were still left out. On the other hand, even with the maximum α value 1, the algorithm failed to detect the half of anomaly points between 11,980 and 13,000 min using the detrended residual series. Such a method divides data points into samplers for labelling anomalies, and this introduces discontinuity for the time series analysis. Therefore, for a collective anomaly like split, it will either only label the outer outliers as anomalous regions or label the entire sequence with a high false positive rate.

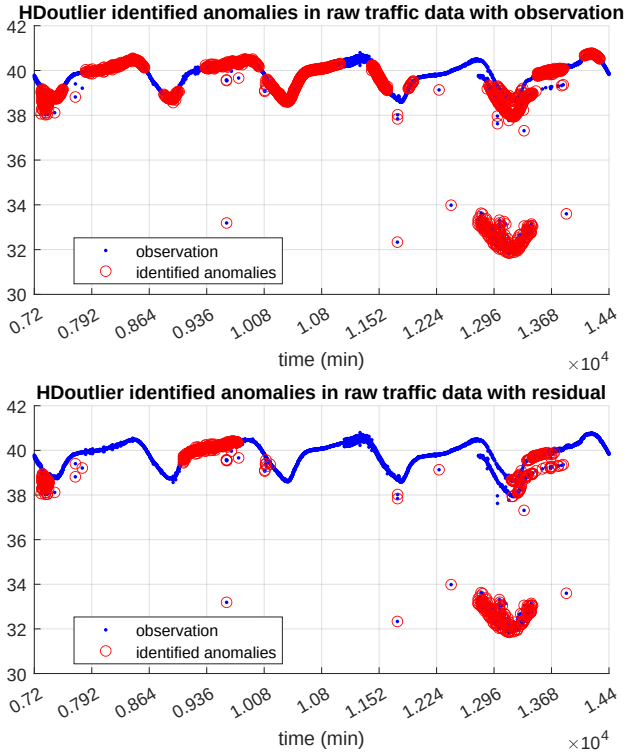


Figure A2. `HDoutlier` anomaly detection on the original BT network traffic data from Day 6 to Day 10.

Appendix B. Type of Anomalies in the Simulated BT Network Traffic Data

Table A1. Type of Anomalies in the Simulated BT Network Traffic Data.

Anomaly Type		Description
Split	1	two parallel levels slightly above normal
	2	three parallel levels below normal
	3	two parallel levels significantly above normal
	4	two parallel levels slightly below normal
	5	three parallel levels above normal
	6	two parallel levels significantly below normal
Interference	7	small intensity with a short period
	8	large intensity with a long period
	9	small intensity with a short decay period
	10	large intensity with a long decay period
Shift	11	short-time small shift above normal
	12	long-time small shift below normal
	13	decay shift above normal
	14	decay shift below normal
	15	short-time large shift above normal
	16	long-time large shift below normal

Table A1. *Cont.*

Anomaly Type		Description
Single point	17	value with small increase (0–0.5)
	18	value with large increase (0–1)
	19	value with small decrease (0–0.5)
	20	value with large decrease (0–1)

References

1. Yu, W.; Li, J.; Bhuiyan, M.Z.A.; Zhang, R.; Huai, J. Ring: Real-time emerging anomaly monitoring system over text streams. *IEEE Trans. Big Data* **2017**, *5*, 506–519. [CrossRef]
2. Rossi, A.; Acito, N.; Diani, M.; Corsini, G. RX architectures for real-time anomaly detection in hyperspectral images. *J. Real-Time Image Process.* **2014**, *9*, 503–517. [CrossRef]
3. Duo, R.; Nie, X.; Yang, N.; Yue, C.; Wang, Y. Anomaly Detection and Attack Classification for Train Real-Time Ethernet. *IEEE Access* **2021**, *9*, 22528–22541. [CrossRef]
4. Nawaratne, R.; Alahakoon, D.; De Silva, D.; Yu, X. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Trans. Ind. Inform.* **2019**, *16*, 393–402. [CrossRef]
5. Song, L.; Liang, H.; Zheng, T. Real-time anomaly detection method for space imager streaming data based on HTM algorithm. In Proceedings of the 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE), Hangzhou, China, 3–5 January 2019; pp. 33–38.
6. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–58. [CrossRef]
7. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [CrossRef]
8. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3768–3774. [CrossRef]
9. Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J.A. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–33. [CrossRef]
10. Aguilera-Martos, I.; García-Barzana, M.; García-Gil, D.; Carrasco, J.; López, D.; Luengo, J.; Herrera, F. Multi-step histogram based outlier scores for unsupervised anomaly detection: ArcelorMittal engineering dataset case of study. *Neurocomputing* **2023**, *544*, 126228. [CrossRef]
11. Billings, S.A. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
12. Fisch, A.; Eckley, I.A.; Fearnhead, P. A linear time method for the detection of point and collective anomalies. *arXiv* **2018**, arXiv:1806.01947.
13. Fisch, A.; Bardwell, L.; Eckley, I.A. Real Time Anomaly Detection And Categorisation. *arXiv* **2020**, arXiv:2009.06670.
14. Fisch, A.T.; Eckley, I.A.; Fearnhead, P. Subset multivariate collective and point anomaly detection. *J. Comput. Graph. Stat.* **2022**, *31*, 574–585. [CrossRef]
15. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [CrossRef]
16. Katser, I. Skoltech Anomaly Benchmark (SKAB). 2021. Available online: https://www.kaggle.com/datasets/yuriykatser/skoltech-anomaly-benchmark-skab (accessed on 24 October 2022).
17. Wei, Y.; Jang-Jaccard, J.; Xu, W.; Sabrina, F.; Camtepe, S.; Boulic, M. LSTM-autoencoder-based anomaly detection for indoor air quality time-series data. *IEEE Sens. J.* **2023**, *23*, 3787–3800. [CrossRef]
18. Jin, M.; Koh, H.Y.; Wen, Q.; Zambon, D.; Alippi, C.; Webb, G.I.; King, I.; Pan, S. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv* **2023**, arXiv:2307.03759.
19. Bardwell, L.; Fearnhead, P. Bayesian detection of abnormal segments in multiple time series. *Bayesian Anal.* **2017**, *12*, 193–218. [CrossRef]
20. James, N.A.; Kejariwal, A.; Matteson, D.S. Leveraging cloud data to mitigate user experience from 'breaking bad'. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 3499–3508.
21. Gu, Y.; Wei, H.L. A robust model structure selection method for small sample size and multiple datasets problems. *Inf. Sci.* **2018**, *451*, 195–209. [CrossRef]
22. Gu, Y.; Wei, H.L.; Boynton, R.J.; Walker, S.N.; Balikhin, M.A. System identification and data-driven forecasting of AE index and prediction uncertainty analysis using a new cloud-NARX model. *J. Geophys. Res. Space Phys.* **2019**, *124*, 248–263. [CrossRef]
23. Gu, Y.; Yang, Y.; Dewald, J.P.; Van der Helm, F.C.; Schouten, A.C.; Wei, H.L. Nonlinear modeling of cortical responses to mechanical wrist perturbations using the narmax method. *IEEE Trans. Biomed. Eng.* **2020**, *68*, 948–958. [CrossRef]
24. Hussain, S.; Al-Alili, A. A new approach for model validation in solar radiation using wavelet, phase and frequency coherence analysis. *Appl. Energy* **2016**, *164*, 639–649. [CrossRef]
25. Mahongo, S.; Deo, M. Using artificial neural networks to forecast monthly and seasonal sea surface temperature anomalies in the western Indian Ocean. *Int. J. Ocean Clim. Syst.* **2013**, *4*, 133–150. [CrossRef]

26. Brentan, B.M.; Campbell, E.; Lima, G.; Manzi, D.; Ayala-Cabrera, D.; Herrera, M.; Montalvo, I.; Izquierdo, J.; Luvizotto, E., Jr. On-line cyber attack detection in water networks through state forecasting and control by pattern recognition. In Proceedings of the World Environmental and Water Resources Congress 2017, Sacramento, CA, USA, 21–25 May 2017; pp. 583–592.
27. Lee, C.C.; Sheridan, S.C. A new approach to modeling temperature-related mortality: Non-linear autoregressive models with exogenous input. *Environ. Res.* **2018**, *164*, 53–64. [CrossRef] [PubMed]
28. Cui, Y.; Bangalore, P.; Tjernberg, L.B. An Anomaly Detection Approach Using Wavelet Transform and Artificial Neural Networks for Condition Monitoring of Wind Turbines' Gearboxes. In Proceedings of the 2018 Power Systems Computation Conference (PSCC), Dublin, Ireland, 11–15 June 2018; pp. 1–7.
29. Bai, M.; Liu, J.; Chai, J.; Zhao, X.; Yu, D. Anomaly detection of gas turbines based on normal pattern extraction. *Appl. Therm. Eng.* **2020**, *166*, 114664. [CrossRef]
30. Taqvi, S.A.; Tufa, L.D.; Zabiri, H.; Maulud, A.S.; Uddin, F. Fault detection in distillation column using NARX neural network. *Neural Comput. Appl.* **2020**, *32*, 3503–3519. [CrossRef]
31. Fisch, A.; Bardwell, L.; Eckley, I.A. *Anomaly: Detecting Anomalies in Data*; R Package Version 4.0.2; R Foundation for Statistical Computing: Vienna, Austria, 2021.
32. *MATLAB*, Version 9.10.0 (R2021a); The MathWorks Inc.: Natick, MA, USA, 2021.
33. Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **1944**, *2*, 164–168. [CrossRef]
34. Hyndman, R.J.; Khandakar, Y. Automatic time series forecasting: The forecast package for R. *J. Stat. Softw.* **2008**, *27*, 1–22. [CrossRef]
35. Bandara, K.; Hyndman, R.J.; Bergmeir, C. MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns. *arXiv* **2021**, arXiv:2107.13462.
36. Wilkinson, L. Visualizing big data outliers through distributed aggregation. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 256–266. [CrossRef] [PubMed]
37. Rand, W.M. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846–850. [CrossRef]
38. Duff, I.S.; Koster, J. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM J. Matrix Anal. Appl.* **2001**, *22*, 973–996. [CrossRef]
39. MacQueen, J. Classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 21 June–18 July 1965; pp. 281–297.
40. Johnson, S.C. Hierarchical clustering schemes. *Psychometrika* **1967**, *32*, 241–254. [CrossRef]
41. Henderson, K.; Gallagher, B.; Eliassi-Rad, T. EP-MEANS: An efficient nonparametric clustering of empirical probability distributions. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, 13–17 April 2015; pp. 893–900.
42. Euán, C.; Ombao, H.; Ortega, J. The hierarchical spectral merger algorithm: A new time series clustering procedure. *J. Classif.* **2018**, *35*, 71–99. [CrossRef]