processes

MDPI

Article

Membrane System-Based Improved Neural Networks for Time-Series Anomaly Detection

Wenxiang Guo [1,2], Xiyu Liu [1,2,*] and Laisheng Xiang [2]

[1] Academy of Management Science, Shandong Normal University, Jinan 250358, China; wenxiang_guo@163.com

[2] Business School, Shandong Normal University, Jinan 250358, China; xls3366@163.com

* Correspondence: xyliu@sdnu.edu.cn

Received: 12 August 2020; Accepted: 12 September 2020; Published: 17 September 2020

Abstract: Anomaly detection in time series has attracted much attention recently and is quite a challenging task. In this paper, a novel deep-learning approach (AL-CNN) that classifies the time series as normal or abnormal with less domain knowledge is proposed. The proposed algorithm combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) to effectively model the spatial and temporal information contained in time-series data, the techniques of Squeeze-and-Excitation are applied to implement the feature recalibration. However, the difficulty of selecting multiple parameters and the long training time of a single model make AL-CNN less effective. To alleviate these challenges, a hybrid dynamic membrane system (HM-AL-CNN) is designed which is a new distributed and parallel computing model. We have performed a detailed evaluation of this proposed approach on three well-known benchmarks including the Yahoo S5 datasets. Experiments show that the proposed method possessed a robust and superior performance than the state-of-the-art methods and improved the average on three used indicators significantly.

Keywords: membrane systems; anomaly detection; time series; convolutional neural networks; long short-term memory

1. Introduction

Anomaly detection aims to find abnormal behavior of data and is widely studied in many fields, like fault detection or predicted maintenance in industrial systems [1]. The reason anomaly detection is important is because anomalies usually contain useful and critical message. To cope with the increasing data collected by research institutions and industries through the Internet of Things (IoT), it is important to have automated procedures that separate the anomalies from normal data.

However, anomaly detection is considered a hard problem [2]. The extremely unbalanced data distribution is the biggest difficulty, and the negative class rate is extremely low. One detection algorithm which works very well on a certain benchmark might get surprisingly bad performance on another. Moreover, anomaly detection for time series is much more difficult due to the issue inherent in time series. For these reasons, this paper tries to find an effective and robust detection algorithm. Many scholars have studied the methods of detecting abnormal patterns by extracting data features in the field of anomaly detection. Anomaly-detection methods mainly consist of three types: statistical modeling [3–6], such as the k-means clustering and Random forest methods, temporal feature modeling [7–10] which is mainly based on the LSTM, and spatial feature modeling [11–13] which takes the advantages of CNN. Traditionally, time-series anomaly detection has been tackled using distance-based methods, such as the dynamic time wrapping algorithm (DTW) [14], meanwhile, artificial neural networks have become powerful tools for time-series anomaly detection due to the large amount of data.

Membrane computing (P system) [15], a novel branch of natural computing, has gained popularity in recent years due to its promising features such as the distribution, uncertainty, and especially, parallelism. The P system were inspired by the structure and function of biological cells and communication in tissues, organs and cell populations [16,17]. Many variants of P system has been proposed and combined with many optimization approaches [18,19] which shows great performance of convergence and robustness [20–24]. Furthermore, with the development of GPU and to make most use of the parallelism of membrane systems, P system has been simulated in GPU [25] recently. However, the common P systems use the simplified membrane structures to deal with problems due to computation purposes; therefore, it is necessary to use complex structures of membranes to solve real applications.

Deep learning has become a popular machine learning approach due to its ability to learn high-level representations related to the data, such as the periodicity and seasonality of the time series. These representations are learned automatically from data with little or no need manual feature engineering and domain expertise [26]. For time-series data, LSTM has become the most widely used model for its ability to learn long-range patterns. LSTM works well in handling the variable-length sequences, but it lacks the ability to extract local contextual information and cannot use the contextual information; therefore CNN is integrated in this paper. Due to these considerations, the main intention of this work is to combine P systems and deep-learning approach to develop a novel framework for time-series anomaly detection. We proposed a hybrid dynamic membrane system (HM-AL-CNN) which reduces the time and takes advantage of ensemble learning in deep P system. In the novel membrane structure, we carry out multiple AL-CNNs for time-series anomaly detection which predicts the label of next timestamp using a window of time-series.

2. Main Contributions

The objective of HM-AL-CNN is to robustly detect time-series point anomalies and discord. As far as we know, this is the first attempt to solve temporal data anomaly-detection tasks via a membrane system-based approach. Profiting from its parallelism, the proposed P system can handle several AL-CNN models with different initialization to get effective features simultaneously. For comparison, we evaluate our methods on three well-known benchmarks that have been employed by many previous approaches. Experimental results show that the proposed methods possess a robust and superior performance compared to the state-of-the-art methods. The following are the main contributions of this paper.

1. A hybrid dynamic P system is proposed to solve complex tasks, which integrate the tree-based and graph-based P system; two types of membrane evolutionary rules are introduced as well.
2. This paper intends to take advantage of the outstanding performance of P systems and deep-learning methods. CNN and LSTM were integrated into the proposed membrane systems for time-series anomaly-detection tasks.
3. The proposed method employs LSTM with attention mechanism; squeeze-and-excitation networks are extended and added to further improve the performance.
4. The proposed approach is evaluated on three well-known benchmarks from different domains and shows better performance than other detection algorithms.

The rest of this paper is arranged as follows. Section 3 gives an overview of the background works and gives the architecture framework of HM-AL-CNN. In Section 4, experimental settings and datasets description are presented. Section 5 provides a detailed evaluation of the proposed algorithm on three well-known benchmarks along with other popular anomaly-detection methods. Finally, conclusions and direction for future work are laid out in Section 6 .

3. Related Works and Method

3.1. Background Works

3.1.1. Long Short-Term Memory and Convolutional Neural Networks

Long short-term memory is a mainstream kind of RNN [27] and is much more complex, capable of learning long-term dependencies. LSTM relieves the problem of vanishing gradient by replacing the self-connected hidden units with memory blocks [28]. LSTM has been adopted widely for machine translation and time-series forecasting. The architecture of an LSTM is shown in Figure 1.

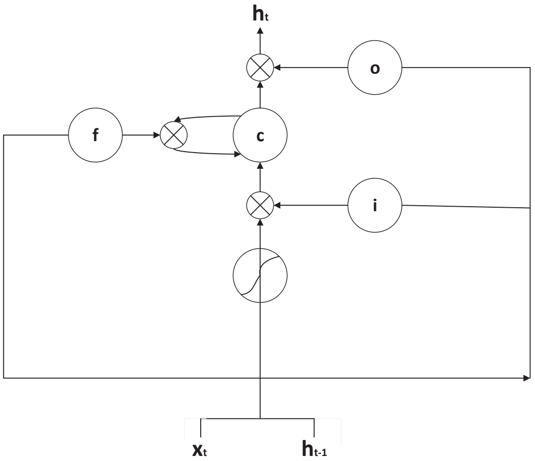


Figure 1. The architecture of an LSTM. i, f, o represents input gate, forget gate and output gate respectively, c indicates the cell activation vector.

The formula of LSTM is given below, the W^*s, R^*s and b^*s are the input weights, the recurrent weights and the biases, respectively, z_t , h_t indicates input and output of the LSTM unit, s_t is the current cell state:

$$z_t = tanh(W^z x_t + R^z h_{t-1} + b^z) \tag{1}$$

$$i_t = \sigma(W^i x_t + R^i h_{t-1} + b^i) \tag{2}$$

$$f_t = \sigma(W^f x_t + R^f h_{t-1} + b^f) \tag{3}$$

$$o_t = \sigma(W^o x_t + R^o h_{t-1} + b^0) \tag{4}$$

$$c_t = z_t(W^o \odot i_t + c_t - 1 \odot f_t) \tag{5}$$

$$h_t = tanh(s_t) \odot o_t \tag{6}$$

Convolutional neural network (CNN) is also a type of ANN and was developed for image classification problems. CNNs can be applied to one-dimensional sequences of data as well, such as human activity recognition; the model can learn an internal representation of the time-series data and achieve comparable performance. The CNN employs a convolution operation and is defined as:

$$(t) = (x * w)(t) \tag{7}$$

This formula can be regarded as a weighted average of $x(\tau)$ at the time stamp t, where weight is calculated by $w(-\tau)$ shifted by amount t. One-dimensional convolutional is defined as:

$$s(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)w(t-\tau) \tag{8}$$

3.1.2. Attention Mechanism

The attention mechanism was proposed by Bahdanau [29] and is used in various deep-learning models. As the function given by the following equation shows, the context vector c_i for the output is calculated using a weighted sum of the annotations h_i which means that the context vector c_i depends on a sequence of annotations $(h_1, ..., h_{T_x})$. Each annotation h_i contains specific information and drops the irrelevant information about the whole input.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{9}$$

where weight α_{ij} is the attention score of each annotation. It can be calculated as follows:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{10}$$

where e_{ij} is the output score of a neural networks given by $a(v_{i-1}, h_j)$, v_{i-1} is the hidden state, h_j indicates the j-th annotation, e_{ij} attempts to capture the alignments of the input at j and output at i.

3.1.3. Tissue-Like and Cell-Like Membrane Systems

In this section, we briefly introduce some concepts related to P systems which are distributed computational parallel models. Membrane computing was inspired by the structure and functions of cells, tissues and organs. In recent years, researchers have turned to the application of membrane computing models. Generally, there are three main families: cell-like P system [30], tissue-like P system [31,32] and neural-like P system [33]. The structure of the tissue-like P system can be viewed as a net, a tissue-like membrane system of degree $m > 0$ is constructed as follows [34]:

$$\Pi = (O, \omega_1, ..., \omega_q, R_1, ..., R_q, R', i_0) \tag{11}$$

where O represents finite non-empty alphabets of objects; $\omega_i (1 \le i \le q)$ are initial multisets of objects present in cell i; R_i are finite sets of evolution rules in cell $i (1 \le i \le q)$, R' is a finite set of communication rules; $i_0 \in \{0, 1, ..., q\}$ indicates the output region where the computation results are placed.

A cell-like P system has a hierarchical arrangement of membranes inside a skin membrane. Each membrane delimits a region where multisets of objects and rules are placed and a set of evolution rules take the form $[\omega \to \omega']$ [34].

3.2. Deep Neural Network with Squeeze-and-Excitation and Attention Mechanism (AL-CNN)

Squeeze-and-Excitation

Hu [35] proposes a squeeze-and-excitation Network (SENet) for CNNs to improve the channel interdependencies. SENet is an architecture for transformation $F_{tr} : X \to U, X \in R^{W' \times H' \times C'}, U \in R^{W \times H \times C}$. We can then represent output of F_{tr} as $U = (u_1, u_2, ..., u_c)$, u_c is defined as follows, $*$ represents the convolution operation and v_c^s is the kernel.

$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s \tag{12}$$

Hu improves the channel interdependencies through squeeze-and-excitation operation. The squeeze process uses a global average pool to get a global understanding of each channel. In our case, similar to the image data, the U is generated by shrinking through the temporal dimension T

to achieve the channel-wise statistics, $z \in R^C$, the cth element of z is calculated by $F_{sq}(u_c)$ which is defined as follows:

$$z_c = F_{sq}(u_c) = \frac{1}{T}\sum_{t=1}^{T} u_c(t) \tag{13}$$

To use the aggregated information obtained from the squeeze stage, excite operation that uses two fully connected layers is employed to get the channel-wise dependencies. We employ an equation given below:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1 z)) \tag{14}$$

where F_{ex} is a neural network, σ and δ indicates the sigmoid function and ReLU function respectively, W_1 and W_2 are learnable parameters of F_{ex}. Finally, the output of the block is gained by rescaling U as follows:

$$\tilde{x}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c \tag{15}$$

where $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_C)$ and $F_{scale}(u_c, s_c)$ represents the channel-wise multiplication between the feature map u_c and the scalar s_c.

Normally, time-series anomaly detection can be transformed into a binary classification problem but is much more complex, for the data is extremely imbalanced. LSTM and CNN have rarely been combined to realize time-series anomaly detection. Similar to the LSTM-FCN proposed by Fazle Karim [36], the model we proposed (AL-CNN) combines both and extends the LSTM with attention mechanism; furthermore, one-dimensional convolution (1DCNN) is added before the attention LSTM to improve the efficiency of the model. In particular, we extend the Squeeze-and-Excitation block to the case of 1D sequence models to enhance the anomaly-detection accuracy. The model can handle both the point anomaly and discords no matter how univariate or multivariate time series. The procedure of the proposed AL-CNN is shown in the Figure 2.

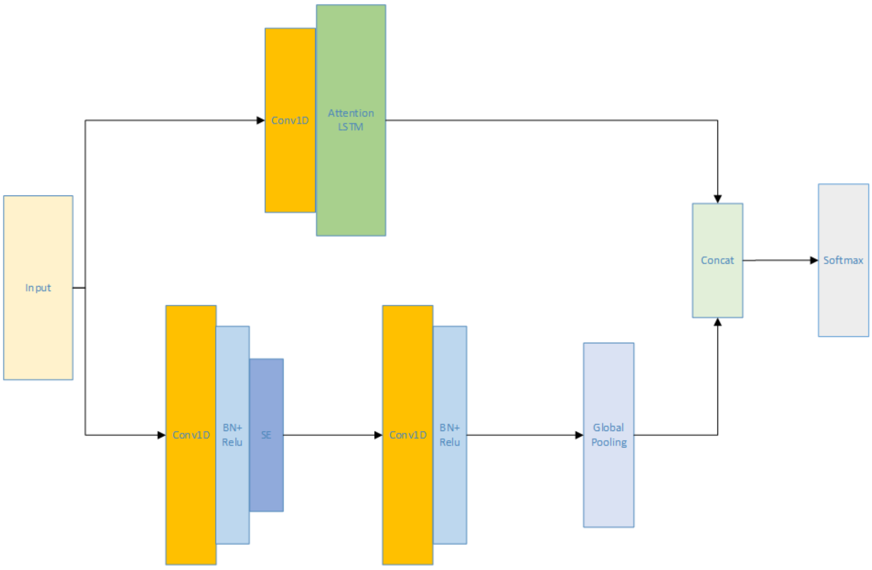


Figure 2. The architecture of the proposed AL-CNN which shows each step of the procedure.

3.3. Hybrid Dynamic Membrane Systems Based Al-Cnn (Hm-Al-Cnn) for Time-Series Anomaly Detection

3.3.1. Architecture Summary of Hm-Al-Cnn

Generally, both tissue-like and cell-like P systems are predigestion and they are not applied to deal with hard problems in the real world. In this work, we intend to use the strengths of both tissue-like and cell-like membrane structure to develop a hybrid dynamic membrane structure as shown in

Figure 3. The graph-based and tree-based membrane structure is depicted via rounded rectangle and squares, respectively. A hybrid dynamic membrane system is constructed as the form:

$$\Pi = (O, E, u, u_T, u_G, \omega_1, ..., \omega_q, R_1, ..., R_q, i_0) \tag{16}$$

where O represents a finite set of objects; $E \subseteq O$ is the set of objects in the environment; μ is a membrane structure which include μ_T and μ_G, here, μ_T are Tree-based membranes and μ_G represent Graph-based membranes; the symbols $\omega_1, ..., \omega_q$ are finite sets of strings over O of q membranes; the i_0 represents the output membrane of Π and $R_1, ..., R_q$ are finite sets of rules including two types described below:

The G-rule is used in the HM-AL-CNN to establish a synchronous communication channel within the computation cells, exchange the multiset a_i of cell x with multiset b_i of cell y, x and y are membrane labels:

$$\Pi = (a_i, out, x; b_i, in, y) \tag{17}$$

The C-rule compares the output of each AL-CNN and picks the best one as the final result:

$$\{pre - Output1, pre - Output2, ..., pre - Outputm\} \rightarrow output \tag{18}$$

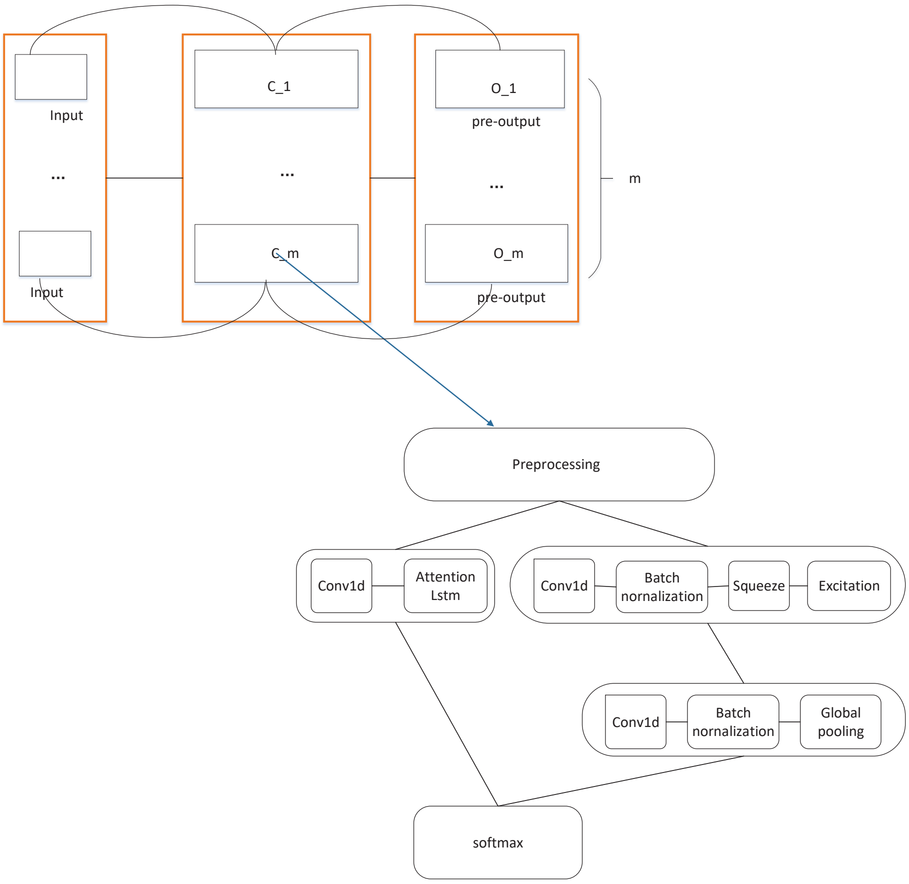


Figure 3. Structure of the proposed P system for time-series anomaly detection.

3.3.2. Initialization

The P system yields the whole initial objects in Input cells; each object represents a 1-dimensional or 2-dimensional vector, denoting the original time series with a size of $w \times h$. Then, the Input cells communicate objects to m computation cells to carry out each AL-CNN.

$$O = (\beta_{11}, \beta_{12}, ...\beta_{1h}; \beta_{i1}, \beta_{i2}, ...\beta_{ih}; \beta_{w1}, \beta_{w2}, ...\beta_{wh}) \tag{19}$$

where $\beta_{i1}, \beta_{i2}, ...\beta_{ih}$ indicates the ith row of the time series.

3.3.3. Computation Mechanism

In this paper, the G-rule is introduced to implement several AL-CNNs and Pre-Output cell gives the result of every AL-CNN. Objects in the P system evolve according to the step of AL-CNN described in Section 3.2 during the computation phase. Then, the C-rule is applied to choose the best objects among the Pre-Outputs as the final result of the HM-AL-CNN.

3.3.4. Termination and Output

The above computing procedures are processed iteratively, and the maximum computation iteration is used as the halting condition. The membrane system halts when the maximum number of iterations is reached and all the objects in the output cell are considered to be the final results of the P system.

4. Experiments

4.1. Experiments Settings

To evaluate the proposed method, HM-AL-CNN has been tested on three benchmarks which are described in Section 4.4. The model was optimized using Adam with an initial learning rate of 1×10^{-6} and the convolution kernels are initialized by the He initialization scheme [37], ReLU was used as the activation function for the hidden layers. The number of training epochs was determined based on the length of the input; for the Yahoo Webscope S5, the model was trained for 500 epochs using batches of 128. The Classic Anomaly Datasets and Space Shuttle Valve Dataset were trained for 700 epochs using batches of 256.

Time-series data need to be transformed into sequences of overlapping windows of size w so that the system makes sense. For x_t at time step t, its condition (normal or abnormal) is used as the label of the former w elements; w is the time window size which is also called a history window. Then, we can define the data as a form of (N, Q, M), where N is the number of samples in the time series, Q indicates the maximum time steps and M represents the number of variables; we define the M to 1 if the time series is univariate.

In addition, both the train and test datasets are normalized using Equation (20). x and x' represents the value of the actual time-series data and the normalized value, respectively. Moreover, we define fixed-sized anomaly windows with each window centered around an anomaly; points in the anomaly window are labeled abnormal. For instance, if the anomaly window size is set to 10, indicating the former 5 points and latter 5 points are labeled abnormal. Only the training sets are operated as such; this up-sampling operation can relieve the extremely imbalance of the data and enhance the performance significantly especially the recall rate.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{20}$$

4.2. Loss Function and Output

Cross-Entropy Loss given in Equation (21) has been employed to measure the difference between the actual value y_j and predicted value $\hat{y}_j$.

$$L = (ylog\hat{y}_j + (1 - y)log(1 - \hat{y}_j)) \tag{21}$$

In our case, the SoftMax layer classifies the output into two classes either normal or abnormal as described in Equation (22). C indicates the class, d is the output of the fully connected layer, w is the weight, L represents the last layer and N_c is the total number of classes.

$$P(c|d) = argmax_{c \in C} \frac{exp(d^{L-1}w^L)}{\sum_{k=1}^{N_c} exp(d^{L-1}w_k)} \tag{22}$$

4.3. Evaluation Metrics

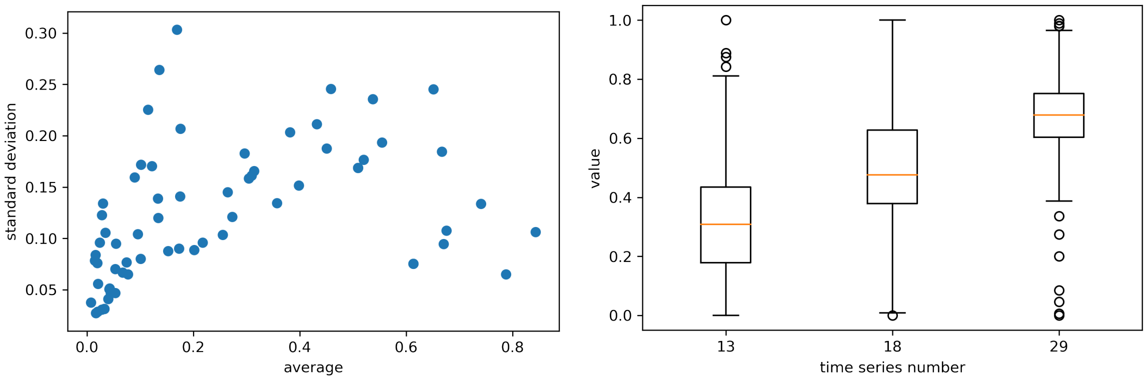
The proposed approach is evaluated using Precision, Recall, F-score and AUC. If an abnormal case is classified as a normal, this type of error is considered to be false negative (FN). True positive (TP), true negative (TN) and false positive (FP) is defined similarly; each algorithm was evaluated through TP, TN, FP and FN rates. In addition, AUC is also the most commonly used metric for evaluating anomaly-detection methods.

4.4. Datasets Description

In this section, we describe three well-known benchmarks from different domains, including real-world and the synthetic datasets which have been applied in previous works on anomaly detection, including the Yahoo Webscope S5, Classic Anomaly-Detection Datasets and Space Shuttle Valve Dataset.

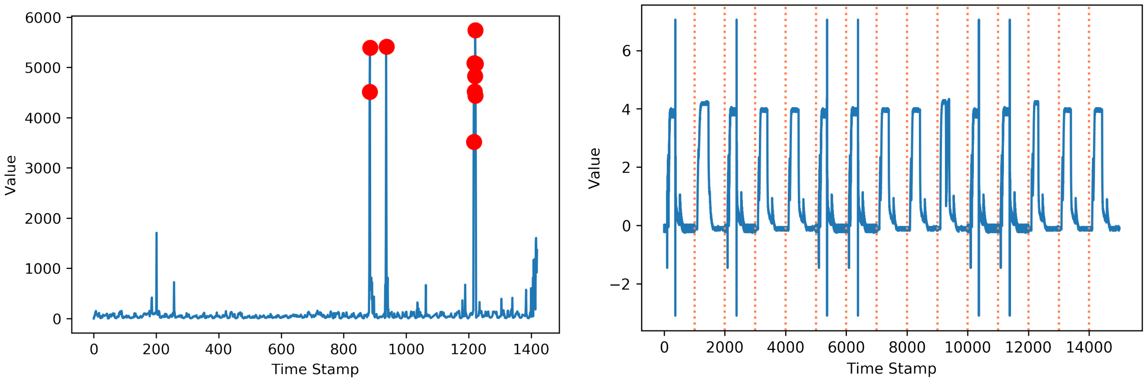
- Yahoo Webscope S5 Datasets

Yahoo Webscope S5 consists of four classes. Class A1 contains the real Yahoo membership login data, and A2, A3 and A4 contain synthetic anomaly data (https://research.yahoo.com). Table 1 shows the characteristics of each sub-benchmark. This dataset contains 367 time series. Each time series consists of almost 1500 data including 0.02% abnormal values. Figure 4a,b show the statistical graphs for the class A1. We can see from the two figures that the data distribution of each file is significantly different; it is not easy to carry out anomaly detection using statistical analysis techniques. Figure 5a shows a real-world time series of the A1 class.



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

Figure 4. (**a**) Standard deviation graph based on the average of the normalized amount of data in each file. (**b**) Distribution of three time series of class A1.



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

Figure 5. (**a**) Example plot of the Yahoo Webscope S5. (**b**) Plot of the NASA Space Shuttle Valve Dataset.

Table 1. Characteristics of each class for the Yahoo Webscope S5.

	A1	A2	A3	A4
Real	√	X	X	X
Synthetic	X	√	√	√
Number of instance	94,866	142,100	168,000	168,000
Number of anomalies	1669	466	943	837

- Classic Anomaly-Detection Datasets

Six commonly used natural datasets have been adopted in this section, which can be found at the UCIRepository [38] and OpenML; anomaly cases have already been marked as ground truth, including the *Pima*, *Covertype*, *Ionosphere*, *Mammography*, *Shuttle* and *Kddcup99*. We have removed all non-continuous attributes as done in [39,40]. Properties of each dataset are shown in Table 2.

Table 2. Characteristics of the six classic datasets.

	Number of Instances	Number of Features	Anomaly Class	Anomaly Rate
Pima	768	8	pos	34.90%
CoverType	286,048	10	class = 4	0.96%
Mammography	11,183	6	class = 1	2%
Ionosphere	351	32	bad	36%
Shuttle	49,097	9	classes $\neq$ 1(class 4 removed)	7%
Kddcup99	494,021	41	class $\neq$ normal	80%

- NASA Space Shuttle Valve Dataset

This dataset collects values which control the flow of fuel on the space shuttle. Some subsequences are normal and few subsequences are abnormal. Figure 5b shows this time series, and the time series is segmented to several subsequences with an orange dotted line; some subsequence are considered abnormal or, in other words, discord subsequences.

4.5. Comparison to State-of-the-Art

Experiments on the Yahoo Webscope S5 are compared to several deep-learning approaches, including the CNN, LSTM, CNN + LSTM, DeepAnt [41] and two popular tools, Yahoo EGADS which was released by Yahoo Labs to detect anomalies in large scale time-series data and Twitter Anomaly-Detection method which aims to detect anomalies of social network data [38]. There are also many different previous works related to the classic anomaly benchmarks mentioned in Section 4.4, for the sake of brevity, we select the popular anomaly-detection techniques for comparison including the Isolation Forest (iForest), OCSVM, LOF [42].

5. Results and Discussion

5.1. Results

- YAHOO Webscope S5 Datasets

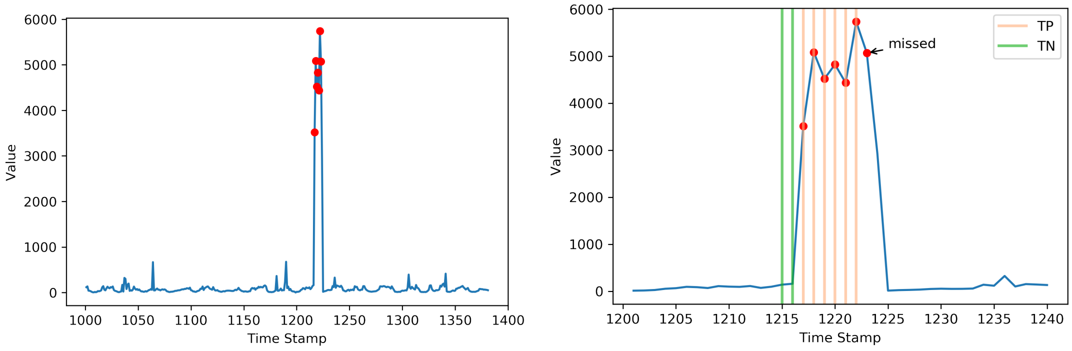
The experimental results of the proposed compared to the other detection algorithms are shown in Table 3 which demonstrates that the proposed improves the detection performance compared to other algorithms, including the deep-learning and classic anomaly-detection algorithms. Figure 6a,b indicate the experimental result of an example time series in the A1 class; HM-AL-CNN detects five out of six anomalies and has only one false positive. Furthermore, HM-AL-CNN does the detection before the true anomalies occur, which is vital to the real application scenery, especially in the industry field. We compared our results with previous methods using the t-test as shown in Table 4. The *p*-values for F-score of A1 are all < 0.05, and the proposed approach achieves a statistically significant improvement over other methods. Table 5 shows a comparison of the proposed with other algorithms on the whole Yahoo Webscope S5. This table gives average F-score of the comparison algorithms along with the proposed data of each sub-benchmark. HM-AL-CNN outperforms other methods in three sub-benchmarks and works slightly worse for sub-benchmark A4. We compared our results with previous methods using a Wilcoxon signed-rank test as shown in Table 6; the proposed approach achieves a statistically significant improvement over other methods except DeepAnt. Even though HM-AL-CNN is not always the best, it achieves better means than DeepAnt and performs better in the whole dataset.

Table 3. Performance comparison of the proposed with other methods on class A1.

Methods	Accuracy	Precision	Recall	F1 Score
Proposed method	0.94	0.59	0.79	0.66
LSTM	0.72	0.14	0.24	0.18
CNN	0.73	0.25	0.50	0.23
CNN + LSTM	0.86	0.38	0.44	0.35
YAHOO EGADS	-	-	-	0.47
Twitter Anomaly Detection	-	-	-	0.48

Table 4. The *p*-values of our method compared to the other methods on class A1.

Methods	Dataset	*p*-Value (F-Score)
LSTM	A1	0.0050 ($p < 0.05$)
CNN	A1	0.0021 ($p < 0.05$)
CNN + LSTM	A1	0.0013 ($p < 0.05$)



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

Figure 6. An example test set result on Yahoo dataset. (**a**) shows the time series; true anomalies are highlighted by red markers. (**b**) shows the detection results. Shading in peach and green denotes detections made by HM-AL-CNN.

Table 5. Performance comparison of the proposed models with other methods on the whole Yahoo Webscope S5.

	A1	A2	A3	A4
Proposed methods	0.66	0.99	0.87	0.59
Yahoo EGADS	0.47	0.58	0.48	0.29
Twitter Anomaly Detection Alpha = 0.0	0.48	0	0.26	0.31
Twitter Anomaly Detection Alpha = 0.1	0.48	0	0.27	0.33
DeepAnt	0.46	0.97	0.87	0.68

Table 6. The p-values of our method compared to the other methods on the whole Yahoo Webscope S5.

Methods	Dataset	p-Value (F-Score)
Yahoo EGADS	Yahoo S5	0.015 ($p < 0.05$)
Twitter Anomaly Detection Alpha = 0.0	Yahoo S5	0.015 ($p < 0.05$)
Twitter Anomaly Detection Alpha = 0.1	Yahoo S5	0.015 ($p < 0.05$)
DeepAnt	Yahoo S5	0.500 ($p > 0.5$)

- Classic Anomaly-Detection Datasets

To evaluate different anomaly-detection algorithms along with the proposed on the Classic Anomaly-Detection Datasets, AUC has been used. AUC is used commonly for evaluating the detection approach on the mentioned datasets. We compare the results of three state-of-the-art anomaly-detection methods with HM-AL-CNN and the results are shown in Table 7. For iForest, OCSVM, and HM-AL-CNN, 40% of the actual data are used for training and rest for testing. We have used the default parameters suggested in [39] for iForest; RBF kernel for OCSVM and $k = 10$ is applied for LOF. Figure 7 shows that HM-AL-CNN has an arithmetic rank of 1.66 and performs better than the existing methods via a critical-difference comparison of the average arithmetic ranks.

Table 7. Performance comparison of the proposed models with the rest methods on Classic Anomaly-Detection Datasets.

	iForest	OCSVM	LOF	DeepAnt	HM-AL-CNN
Pima	0.4	0.26	0.48	0.31	0.65
ForestType	0.78	0.70	0.57	0.85	0.92
Ionosphere	0.82	0.84	0.83	0.85	0.86
Mammography	0.85	0.89	0.72	0.99	0.97
Shuttle	0.98	0.99	0.56	0.99	0.94
kddcup99	0.98	0.99	0.42	0.99	0.99

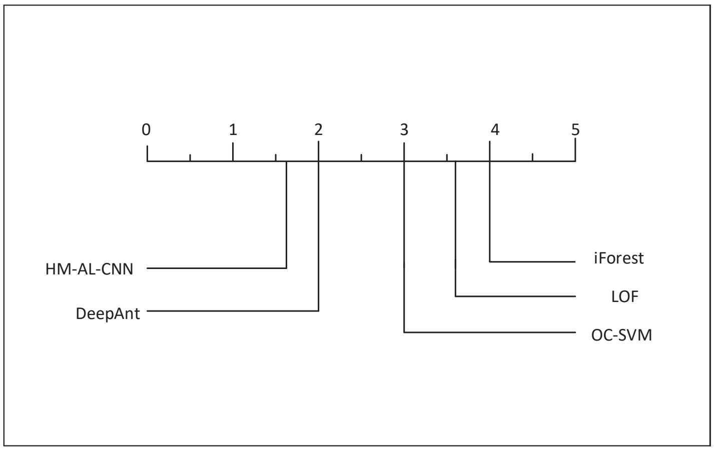
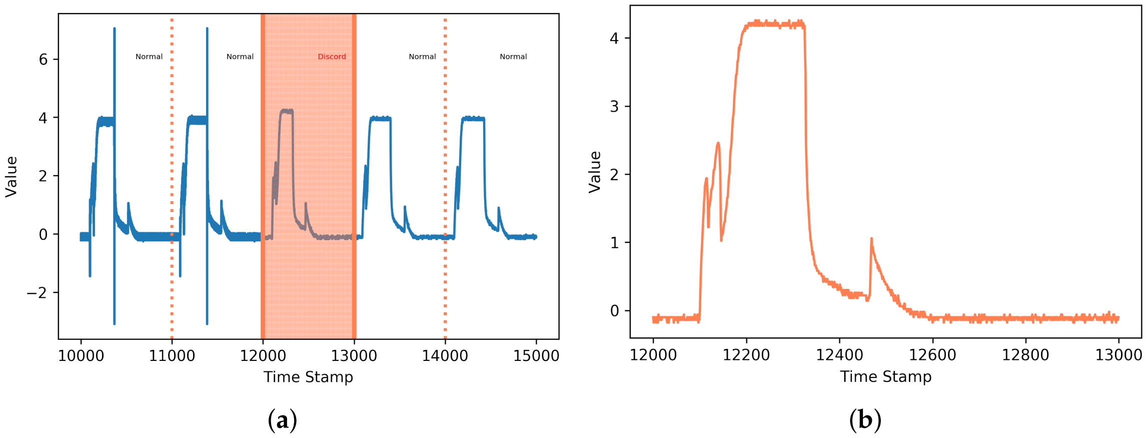


Figure 7. Critical difference of the arithmetic means of the ranks on six datasets.

- NASA Space Shuttle Valve Dataset

The above experiments have already shown that HM-AL-CNN is able to detect point anomalies in time-series data. In this section, HM-AL-CNN is proved to be suitable to time-series discord detection as well. Discords are subsequences that are different from the rest of a longer time series [43]. In this experiment, this proposed algorithm can label most of the points in an abnormal discord cycle and label the points in normal cycles normally. If the abnormal number is over the threshold we set, we classify the sequence discord. Figure 8a,b show the experiment results, there are four normal sequences and one discord in the test set; these experimental results demonstrate that the proposed algorithms work well.



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

Figure 8. Detection results of the proposed methods.As shown in (**a**), subsequence labeled in orange color refers to a discord, shaded in orange denotes detections made by the proposed. The discord sequence detected by HM-AL-CNN is shown specifically in (**b**).

5.2. Discussion

The proposed method achieves better results than the other methods in most cases. Due to the different distribution of the time series, the proposed works slightly worse in some cases. From Table 3, we found that the proposed performs better than CNN and LSTM-based algorithms, which indicates that squeeze-and-excitation and attention mechanism could be used in this case to improve the detection performance; in addition, the proposed achieves a recall value of 0.79, which improves the previous methods significantly. Table 5 shows that the proposed works slightly worse in A4. Whether or not the additional attributions such as the change-point and noise caused the slightly bad performance still need to be explored, generally, the proposed works better on the whole Yahoo S5. Table 7 indicates that HM-AL-CNN has an arithmetic rank of 1.66 and can find anomalies in a multi-variant dataset as well.

6. Conclusions

In this paper, we propose a novel hybrid dynamic membrane system which takes advantages of tissue-like and cell-like P system for a time-series anomaly-detection task. To get more accurate detection results, CNN and LSTM with attention mechanism are combined, and 1D Squeeze-and-Excitation mechanism is introduced to better learn effective features. Two types of rules are introduced in the designed membrane system, profiting from the parallelism of P system; this proposed HM-AL-CNN can process several AL-CNN models individually, which consumes less time. Experiments show that the proposed possesses better performance than other time-series anomaly-detection algorithms in different benchmarks. However, there are still many important parameters that need to be chosen manually in our system, which remains to be addressed. Evolutionary algorithms such as the particle swarm optimization could be used in the future. Moreover, the design of a more effective membrane system to solve complex problems is also meaningful.

Author Contributions: Conceptualization, W.G. and X.L.; methodology, W.G.; formal analysis, W.G.; resources, W.G., X.L. and L.X.; writing–original draft preparation, W.G.; writing–review and editing, W.G. and X.L.; supervision, X.L.; funding acquisition, X.Y. and L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research project was partially supported by the National Natural Science Foundation of China (61876101, 61802234, 61806114), Natural Science Foundation of Shandong Province, China (ZR2019QF007). The Ministry of Education Humanities and Social Science Research Youth Foundation, China (19YJCZH244), Social Science Fund Project of Shandong Province, China (16BGLJ06, 11CGLJ22), Special Postdoctoral Project of China (2019T120607) and Postdoctoral Project of China (2017M612339,2018M642695).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Varun, C.; Arindam, B.; Vipin, K. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 1–58.
2. Campos, G.; Zimek, A.; Sander, J. On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical Study. *Data Min. Knowl. Discov.* **2016**, *30*, 891–927. [CrossRef]
3. Alizadeh, H.; Khoshrou, A.; Zuquete, A. Traffic classification and verification using unsupervised learning of Gaussian Mixture Models. In Proceedings of the 2015 IEEE International Workshop on Measurements and Networking, Coimbra, Portugal, 12–13 October 2015; pp. 1–6.
4. Munz, G.; Li, S; Carle, G. Traffic anomaly detection using k-means clustering. In Proceedings of the GI/ITG Workshop MMBnet, Hamburg, Germany, 14 September 2007; pp. 13–14.
5. Zhang, J.; Zulkernine, M. Anomaly based network intrusion detection with unsupervised outlier detection. In Proceedings of the 2006 IEEE International Conference on Communications, Istanbul, Turkey, 11–15 June 2006; pp. 2388–2393.
6. Moore, A.W.; Zuev, D. Internet traffic classification using bayesian analysis techniques. *ACM Sigmetrics Perform. Eval. Rev.* **2005**, *33*, 50–60. [CrossRef]
7. Cheng, M.; Li, Q.; Lv, J. Multi-Scale LSTM model for BGP anomaly classification. *IEEE Trans. Serv. Comput.* **2018**. [CrossRef]
8. Ding, Q.; Li, Z.; Batta, P. Detecting BGP anomalies using machine learning techniques. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 003352–003355.
9. Malhotra, P.; Vig, L.; Shroff, G. *Long Short Term Memory Networks for Anomaly Detection in Time Series*; Presses Universitaires de Louvain: Louvain la Neuve, Belgium, 2015; Volume 89.
10. Chauhan S.; Vig, L. Anomaly detection in ECG time signals via deep long short-term memory networks. In Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 19–21 October 2015; pp. 1–7.
11. Wang, W.; Zhu, M.; Wang, J. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 43–48.
12. Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; Zhao, J.L. Time series classification using multi-channels deep convolutional neural networks. In Proceedings of the International Conference on Web-Age Information Management, Macau, China, 16–18 June 2014; pp. 298–310.
13. Ren, Y.; Wu, Y. Convolutional deep belief networks for feature extraction of EEG signal. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2014; pp. 2850–2853.
14. Felzenszwalb, P.F.; Zabih, R. Dynamic programming and graph algorithms in computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 721–740. [CrossRef]
15. Paun, G. Computing with membranes. *Comput. Syst.* **2000**, *61*, 108–143. [CrossRef]
16. Paun, G.; Rozenberg, G.; Salomaa, A. Membrane computing: Brief introduction, recent results and applications. *Biosystem* **2006**, *85*, 11–12. [CrossRef]

17. Paun, G.; Perez-Jimenez, M.J. *The Oxford Handbook of Membrance Computing*; Oxford University: Oxford, UK, 2010.
18. Lu, K.; Zhou, W.; Zeng, G.; Zheng, Y. Constrained population extremal optimization-based robust load frequency control of multi-area interconnected power system. *Int. J. Electr. Power Energy Syst.* **2019**, *105*, 249–271. [CrossRef]
19. Zeng, G.Q.; Chen, J.; Dai, Y.X.; Li, L.M.; Zheng, C.W.; Chen, M.R. Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. *Neurocomputing* **2006**, *160*, 173–184. [CrossRef]
20. Peng, H.; Shi, P.; Wang, J.; Riscos-Nunez, A.; Perez-Jimenez, M.J. Multiobjective fuzzy clustering approach based on tissue-like membrane systems. *Knowl. Based Syst.* **2017**, *125*, 74–82. [CrossRef]
21. Zhang, G.; Cheng, J.; Gheorghe, M.; Meng, Q. A hybrid approach based ondifferential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems. Multiobjective fuzzy clustering approach based on tissue-like membrane systems. *Appl. Soft Comput.* **2013**, *13*, 1528–1542. [CrossRef]
22. Huang, L.; Suh, I.H.; Abraham, A. Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. *Inform. Sci.* **2011**, *181*, 2370–2391. [CrossRef]
23. Xue, J.; Camino, A.; Bailey, S.T.; Liu, X.; Li, D.; Jia, Y. Automatic quantification of choroidal neovascularization lesion area on oct angiography based on density cell-like p systems with active membranes. *Biomed. Opt. Express* **2018**, *9*, 3208–3219. [CrossRef] [PubMed]
24. Peng, H.; Jiang, Y.; Wang, J.; Perez-Jimenez, M. Membrane clustering algorithm with hybrid evolutionary mechanisms. *J. Softw.* **2015**, *26*, 1001–1012.
25. Muniyandi, R.C.; Sundararajan, E. Using graphics processing unit to accelerate simulation of membrane computing. In Proceedings of the 2017 6th International Conference on Electrical Engineering and Informatics, Langkawi, Malaysia, 25–27 November 2017; pp. 1–6.
26. Chakraborty, D.; Narayanan, V.; Ghosh, A. Integration of deep feature extraction and ensemble learning for outlier detection. *Pattern Recognit.* **2019**, *89*, 161–171. [CrossRef]
27. Pascanu, R.; Gulcehre, C.; Cho, K. How to construct deep recurrent neural networks. *arXiv* **2013**, arXiv:1312.6026.
28. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
29. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
30. Pan, L.; Wu, T.; Su, Y. Cell-like spiking neural P systems with request rules. *IEEE Trans. Nanobiosci.* **2017**, *16*, 1. [CrossRef] [PubMed]
31. Xue, J.; Yan, S.; Wang, Y.; Liu, T.; Qi, F.; Zhang, H.; Qiu, C.; Qu, J.; Liu, X.; Li, D. Unsupervised segmentation of choroidal neovascularization for optical coherence tomography angiography by grid tissue-like membrane systems. *IEEE Access* **2019**, *7*, 143058–143066. [CrossRef]
32. Song, B.; Pan, L.; Perez-Jimenez, M.J. Cell-Like P Systems With Channel States and Symport/Antiport Rules. *IEEE Trans. Nanobiosci.* **2016**, *15*, 555–566. [CrossRef] [PubMed]
33. Zhang, X.; Liu, Y.; Luo, B.; Pan, L. Computational power of tissue P systems for generating control languages. *Inform.Sci.* **2014**, *278*, 285–297. [CrossRef]
34. Xue, J.; Yan, S.; Qu, J.; Qi, F.; Liu, X. Deep membrane systems for multitask segmentation in diabetic retinopathy. *Knowl. Based Syst.* **2019**, *183*, 104887. [CrossRef]
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
36. Karim, F.; Majumdar, S.; Darabi, H. LSTM fully convolutional networks for time series classification. *IEEE Access* **2018**, *6*, 1662–1669. [CrossRef]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. UDelving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago de Chile, Chile, 7–13 December 2015; pp. 1026–1034.
38. Asuncion, A.; Newman, D. *UCI Machine Learning Repository*; UCI: Irvine, CA, USA, 2007.
39. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422.
40. Goix, N. How to evaluate the quality of unsupervised anomaly detection algorithms. *arXiv* **2015**, arXiv:1607.01152.

41. Munir, M.; Siddiqui, S.A.; Dengel, A. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access* **2018**, *7*, 1991–2005. [CrossRef]
42. Breunig, M.M.; Kriegel, H.P.; Ng, R.T. LOF: Identifying density-based local outliers. *ACM SIGMOD Rec.* **2000**, *29*, 93–104. [CrossRef]
43. Keogh, E.; Lin, J.; Fu, A. Hot sax: Efficiently finding the most unusual time series subsequence. In Proceedings of the Fifth IEEE International Conference on Data Mining, Houston, TX, USA, 27–30 November 2005; p. 8.