water

MDPI

Article

Detecting Pattern Anomalies in Hydrological Time Series with Weighted Probabilistic Suffix Trees

Yufeng Yu [1,*], Dingsheng Wan [1], Qun Zhao [1,2] and Huan Liu [2]

[1] College of Computer and Information, Hohai University, Nanjing 210098, China; dshwan@hhu.edu.cn (D.W.); qzhao47@asu.edu (Q.Z.)

[2] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA; Huan.liu@asu.edu

* Correspondence: yfyu@hhu.edu.cn; Tel.: +86-139-5167-0067

Received: 29 March 2020; Accepted: 19 May 2020; Published: 21 May 2020

Abstract: Anomalous patterns are common phenomena in time series datasets. The presence of anomalous patterns in hydrological data may represent some anomalous hydrometeorological events that are significantly different from others and induce a bias in the decision-making process related to design, operation and management of water resources. Hence, it is necessary to extract those "anomalous" knowledge that can provide valuable and useful information for future hydrological analysis and forecasting from hydrological data. This paper focuses on the problem of detecting anomalous patterns from hydrological time series data, and proposes an effective and accurate anomalous pattern detection approach, *TFSAX_wPST*, which combines the advantages of the Trend Feature Symbolic Aggregate approximation (*TFSAX*) and weighted Probabilistic Suffix Tree (*wPST*). Experiments with different hydrological real-world time series are reported, and the results indicate that the proposed methods are fast and can correctly detect anomalous patterns for hydrological time series analysis, and thus promote the deep analysis and continuous utilization of hydrological time series data.

Keywords: hydrological time series; anomalous pattern detection; Variable Markov Model; *TFSAX*; weighted Probabilistic Suffix Tree

1. Introduction

In the era of Big Data, new satellite, space, airborne, shipborne and ground-based remote sensing systems, as well as Internet of Things (IoT) devices, are ubiquitous, producing data rapidly and continuously, which lead to hydrological time series being acquired at a breathless pace, both in size and variety [1,2]. However, due to measurement/manual operation errors, instrument failure, changes in natural laws caused by human activities or hydrological evolution, there is a large number of "anomalous" data in hydrological time series. Undoubtedly, those "anomalous" data will significantly affect the models related to flood forecasting and hydrological analysis, and lead to potentially incomplete or inaccurate results [3]. Therefore, detecting those "anomalies" in hydrological datasets is becoming an important and urgent task for hydrology and information researchers [4].

Anomalies are individuals that behave in an unexpected way or feature abnormal properties [5]. According to the literature [6,7], anomalies in time series can be divided into point anomalies and pattern anomalies, and the problem of finding those unexpected individual points or patterns is referred to as anomaly detection. For hydrological time series data, many researchers have proposed different anomaly detection algorithms from different application aspects to address "anomalous" variables [8–15]. However, those methods pay more attention to detect point anomalies to improve

hydrological data quality rather than mine potentially meaningful pattern anomalies within a given time series.

Pattern anomalies in hydrologic time series may be related to disastrous hydrometeorological events (flood or drought) within a period of time [16]. Therefore, detecting and analyzing pattern anomalies in hydrological time series is helpful to discover the law of a hydrological process, to provide decision support for early warning and prevention of flood and drought disasters, and to reduce economic and social losses. However, there are very few studies focusing on hydrological time series pattern anomaly detection. Moreover, due to the fact that the nature of the time series and anomalies are fundamentally divergent in different domains, it is hard to apply those pattern anomaly detection algorithms that are effective in other areas [17–22] to the hydrology field.

Therefore, this paper proposes a novel pattern anomaly detection algorithm, *TFSAX_wPST*, to detect hydrological time series pattern anomalies. The algorithm first uses the Trend Feature Symbolic Aggregate approximation (*TFSAX*) [23] to discretize original time series into symbolic time series, then proposes the weighted Probability Suffix Tree (*wPST*) to construct the symbol sequence obtained by the above steps, and thus *top-k* pattern anomalies are analyzed and verified from the candidate pattern anomalies set based on the sequence that was pruned during the *wPST* construction process. Experimental results show *TFSAX_wPST* can accurately detect pattern anomalies in hydrological time series and thus provides technical and application support for hydrological time series data analysis and decision-making.

2. Related Work

2.1. Time Series Pattern Anomaly Detection

A time series pattern anomaly represents a pattern with anomalous behavior that is significantly different from other patterns within a given time series. Generally, a pattern may contain a collection of data instances, where each single data instance is not anomalous; however, the combination of them may be an anomaly and implies more important information [6]. For example, it may be a normal condition if the mean daily water level of a station on some day of July is lower than its mean water level over the same period in history. But it may indicate an anomalous pattern (drought event) when the mean daily water level of all 31 days in July at this station is lower than its mean water level over the same period in history. Therefore, detecting and analyzing pattern anomalies that contain more interesting information is more meaningful and valuable [6].

A time series pattern (*TSP*) represents a certain characteristic trend within a given time series, which may be a statistical characteristics metric (e.g., maximum, minimum or mean values of a segment) or mathematical transformation (e.g., Fourier transform). Formally, given a time series *TS*, the pattern of *TS* can be formally represented as a pattern–time tuple:

$$TS = <(m_1, t_1), (m_2, t_2) \ldots (m_i, t_i) \ldots (m_N, t_N)> \; i = 1,2 \ldots N \quad (1)$$

where tuple (m_i, t_i) indicates that the pattern of *TS* is m_1 during 0–t_1, m_2 during t_1–t_2, m_N during t_{N-1}–t_N, and so on and so forth.

Researchers use *Novelty Pattern, Surprise Pattern, Discord, Novel Event* and *Aberrant Behavior* to describe anomaly patterns, and thus design window-based [6,24], similarity-based [25,26], symbolic representation-based [21,27] and model-based [28–31] algorithms to detect the tuples (m_i, t_i) on a given time series *TS* that met the definition of the pattern anomaly based on different application areas and purposes [32].

Wan [33] proposed the *FP_SAX (*Feature Points Symbolic Aggregate Approximation) approach to improve the selection of feature points, and then detect those patterns that have the top-k distance measured by Symbol Distance-based Dynamic Time Warping (*SD_DTW*) as anomalous patterns on hydrological time series. Zhang [34] proposed the distance-based anomalous patterns detection method by improving the selection of feature points in *FP_SAX* [33] and its distance measurement method.

Those approaches have a lower fitting error and higher accuracy in the task of anomalous hydrological time series patterns mining, but how to choose a proper number (N) of feature points and the anomaly determination threshold will affect the detection accuracy and results of the algorithm. Wu et.al [35] used the quantile perturbation method (QPM) to reveal rainfall time series anomalies and changes over the Yellow River Basin due to the fragile ecosystem and rainfall-related disasters. The QPM method is a tool for analyzing extreme values and effective for the identification and analysis of extreme meteorological events. However, it is relatively weak for the detection of ordinary anomalous events.

For detecting pattern anomalies in time series, some existing methods in the literature reduce the problem to a point anomaly detection problem before solving it [11]. In some other methods, pattern anomalies are detected by using different machine learning and data mining approaches [3]. Markov Models (MM) [36] and their variants [29,31] are the popular machine learning approaches extensively used for pattern anomalies detection in time series. In the next section, we briefly study *PST*-based anomaly detection approaches.

2.2. PST-Based Anomaly Detection

The Markov model is a powerful finite state machine and widely used in sequence modeling. The Markov approaches are used in several studies to solve anomaly detection problems with the idea that an odd behavior might be represented not only by a single observation, but also by a series of consecutive observations [36]. The Probabilistic Suffix Tree (*PST*) is a compact representation of the Variable Order Markov Model (VMM) and uses a suffix tree as its storage structure. It originally comes from Probabilistic Suffix Automata (PSA) [37] and is believed to have a more memory efficient representation than the PSA. Hence, it has been used in several domains as an efficient approach for classifying sequences [38,39].

Figure 1 is an example of a *PST* corresponding to string s_1: *abbabbabaaba* over the alphabet $\Sigma = \{a, b\}$ and tree depth $L = 2$. In *PST*, each edge is labelled by a unique symbol σ in Σ. Each node has at most two ($|\Sigma|$) children and records a string representing a path from the node to the root. The node also records a probability distribution vector corresponding to the conditional probabilities of seeing a symbol right after the label string in the dataset [40]. *PST* models the normal behave using the maximum likelihood criterion likelihood ratio. For a given sequence S and its *PST* T, the total likelihood-ratio of the observations can be expressed mathematically as $L = Pr(S|T)$. If the probability of the observation sequence given the model has the largest likelihood ratio (or exceeding a certain preset threshold θ), then an anomaly is detected [29,41].

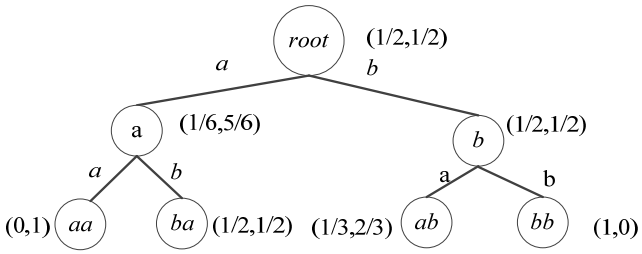

Figure 1. Probabilistic Suffix Tree (*PST*) representation for "*abbabbabaaba*".

However, *PST* is a sequence statistical model based on VMM. It inherits the shortcomings, such as losing important sequence information and reducing detection accuracy of the VMM in anomaly detection tasks [42]. For subsequence A: *abbabbabaaba*, and B: *abbabaaaaaaa*, the probability that seeing a right after event ab in subsequence A ($P_A(a|ab)$) is equal to that of subsequence B ($P_B(a|ab)$). However, the frequency of event ab occurring in A ($P_A(ab) = 4/11$) is higher than that in B ($P_B(ab) = 2/11$). Hence, if it only uses probability to represent the sequence for anomaly detection tasks, it may lead to an erroneous analysis result.

Therefore, we propose a novel *wPST* model to better descript and accurately distinguish different time series sequences; thus, we give here a formal definition for pattern anomaly based on *wPST* to define the detection boundary and detection target for our anomaly detection algorithm.

3. A Novel Time Series Anomaly Detection Approach *TFSAX_wPST*

In this section, we propose a novel time series pattern anomaly detection approach, *TFSAX_ wPST*. Firstly, we propose a novel *wPST* model as the structure to store symbol sequences and give a formal definition for a hydrological time series pattern anomaly based on *wPST*. Then, we conduct a novel time series anomaly detection approach *TFSAX_wPST* to detect pattern anomalies within a given hydrological time series. *TFSAX_wPST* can be performed as in Figure 2.

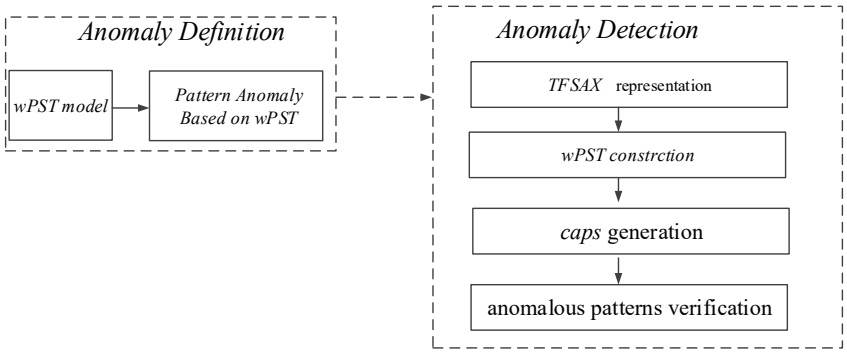

Figure 2. Framework of the Trend Feature Symbolic Aggregate approximation and weighted *PST* approach (*TFSAX_wPST*).

(1) Anomaly Definition: As the basis of anomaly detection, the anomaly definition determines the object of the detection algorithm, the accuracy and interpretability of detection results. Therefore, we give a pattern anomaly definition based on our *wPST* model.

(2) Anomaly Detection: Based on the *wPST* model and our previous research work, *TFSAX*, we propose a novel *TFSAX_wPST* algorithm to detect those patterns that meet our definition within given time series.

3.1. Time Series Pattern Anomaly Based wPST Model

The *wPST* model is an improvement of the *PST* model. It increases the model frequency weight of the subsequence corresponding to a node to distinguish different sequences accurately. For a given sequence, its *wPST* model can be defined as follows:

$$P^T(s) = P^T(\sigma_i|\sigma_1\sigma_2 \ldots \sigma_{i-1}) \times w_i \quad (2)$$

where w_i is the frequency weighting of the subsequence $\sigma_1, \sigma_2 \ldots \sigma_{i-1}$.

Figure 3 shows the *wPST* model of the sequence s_1 in Figure 1. Compared to *PST*, each node in the figure stores the conditional probability distribution vector of the subsequent symbol as well as the frequency weight corresponding subsequence, and thus can better present the feature information of the sequence.

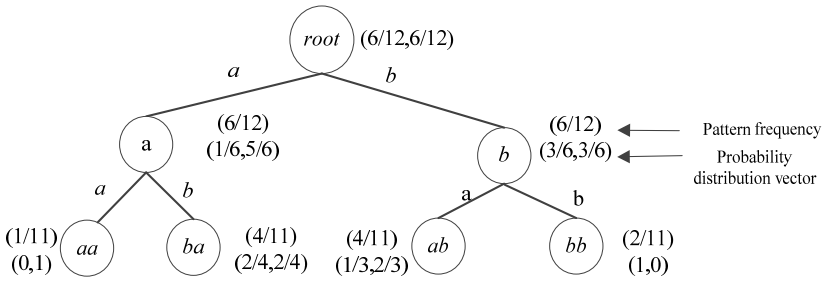

Figure 3. *wPST* representation for "*abbabbabaaba*".

The representation of a *TSP* is a trend feature over a period of time, such as a rising, a stable or a falling subsequence. Moreover, a time series *TS* of length *n* can be treated as a plurality of subsequences of length *m* (*m* <<= *n*); each subsequence has its own trend feature as well as the overall trend features of the original sequence. Therefore, the pattern features between different subsequences are closely related to the temporal trend of the sequence. For these reasons, this paper gives the definition of the hydrological time series anomaly subsequence and time series pattern anomalies based on the physical mechanism as below.

Definition 1. *Time series anomalous subsequence.*

Given a time series *TS*, the event sequence set is Σ, the subsequence *s*, and the subsequent events σ of *s* ($\sigma = suffix(s)\ \&\ \sigma \in \Sigma$). Let Pr_{min} and *MinCt* represent the predefined minimum occurrence probability and minimum occurrence number for the conditional occurrence probability of σ under the condition of *s*, respectively. If the conditional occurrence probability of σ under the condition of *s* is satisfied:

(1) $Pr(\sigma|s) < Pr_{min}$ and
(2) $occ\_num(\sigma) \leq MinCt$,

then, it can define sσ to be an anomalous subsequence on time series *TS*.

As it can be seen from Definition (1), for a given time series *TS*, the smaller the probability of $s\sigma$ occurring under the condition of *s* occurring, the higher the anomaly probability of $s\sigma$ is. Therefore, the top *k* subsequences with the smallest occurrence probability (or occurrence number) are the *top_k* anomaly subsequences.

Definition 2. *Time series anomaly pattern.*

The time series pattern anomaly is a pattern consisting of one or a series of consecutive anomalous subsequences within a given time series.

3.2. TFSAX_wPST Algorithm

3.2.1. *TFSAX* Representation

TFSAX is our previous work to extend the *SAX* representation. *TFSAX* employs the sequence mean feature and trend feature to represent the time series, and thus overcomes the shortcomings of *SAX* that only uses the mean values to describe the original time series. It can be obtained as follows:

Step 1: Normalization. Transform the original time series TS into the normalized time series TS' with a mean of 0 and standard deviation of 1.

Step 2: Dimensionality reduction. Use the *PAA* (Piecewise Aggregate Approximation) approach [43] to divide time series TS' into w equal-sized segments, then extract the mean feature and trend feature of each segment.

Step 3: Discretization. According to the breakpoints lookup table, choose alphabet cardinality, obtain the Trend Feature Symbolic Representation of the original time series and discretize TS′ into symbols, denoted by TS.

For more detailed information about TFSAX, please refer to our previous research work [23].

3.2.2. *wPST* Construction

The construction of *wPST* starts from a subsequence with a single element. It first initializes an empty *wPST* containing only one root node, and then iterates through all possible subsequences, with the length varying from 1 to *L*. For each to be checked subsequence s, its occurrence time should be greater than *MinCt*; then, continue to search the subsequent symbol σ ($\sigma \in \Sigma$) of *s* and count the occurrence times of sσ and σs to determine whether σ needs to be constructed in *wPST*; otherwise, discriminate all the subsequences, beginning with *s*, as being pattern anomalies, and stops searching.

To improve the algorithm's efficiency, this approach uses the hash map data structure at each level to search and update the information of each node before and after a segment in the sequential database. For example, assume that we are at level 2 and the alphabet is {*a*, *b*}. Then, without pruning, the hash.keys at level 2 are all the possible 2 combinations of the alphabet:{*aa*, *ab*, *ba*, *bb*}. These combinations are lexicographically ordered, and the orders are stored at hash.values. Thus, the hash.values are {0, 1, 2, 3}. Now, the hash.key, hash.value combination is used as an index to the arrays A_{before} and A_{after}. Moreover, The arrays' size is the size of the alphabet, and the value of each element of A_{before} is the current count of $\sigma s'$, where s' is the element of hash.keys and σ is a character in the alphabet. Similarly, A_{after} will store the count of $s'\sigma$. Thus, we can update all the counts at each level of the tree after one scan. After a level of the *wPST* is constructed, the current hash map is destroyed and a new hash map for the next level is initialized. For example, assuming that we have a sequential database consisting of one sequence {*abba*}, in one scan we can update the counts of *ab*→ b, *a*← *bb*, *bb*→ *a* and *b*← *ba*. The formal description of constructing *wPST* is shown in Algorithm 1.

Algorithm 1 *wPST* construction. *Build_wPST*(S,H)

Input: Sequence *S*, Maximum depth *H*
Output: *wPST T*

Initialize: $T \leftarrow root$; $k = 0$;
  $k = 1$, $S_1 \leftarrow \{\sigma \mid \sigma \in \Sigma \wedge occ\_count(\sigma) > 0\}$
  $HM_1 \leftarrow HASHMAP(S_1)$
While $k \leq H$ ***Do***
Foreach $(s' \in S_k)$
    $A_{before}[|\Sigma|], A_{after}[|\Sigma|] \leftarrow 0$;
For $i = 1$ *to* $len(s) - k + 1$
  ForEach $(s_{[i,i+k-1]} \in S)$
    If $s_{[i,i+k-1]} \in HM_k.keys$ *then*
      $Update(occ\_times(s_{[i,i+k-1]}))$;
  ForEach$(\sigma \in \Sigma \mid \sigma' \in \Sigma)$
    If $(s[i+k] = \sigma)$ *then* $Update(A_{after}(s[i+k]))$;
    If $(s[i-1] = \sigma')$ *then* $Update(A_{before}(s[i-1]))$;
  ForEach $(s' \in S_k)$
    $T.Add(represent(u, s'))$;
    $w(represent(u, s')) = occ\_times(s')/(len(S)-k)$;
  ForEach $(\sigma \in \Sigma)$
    compute $Pr(\sigma|s')$ *using* A_{after};
    smooth $Pr(\sigma|s')$;
    Mine_candidate_Anomaly $(T, MinCt, Pr_{min})$;
 $HM_{k+1} \leftarrow HASHMAP(S_{k+1})$;
 Return T

3.2.3. Candidate Anomalies Pattern Set Generation

Theoretically, the number of entries in the hash map on the *L*th level *of wPST* is $|\Sigma|^{L-1}$ without pruning while *wPST* is constructed. Therefore, the total complexity of this implementation is *O(NmL)* + $O(L \times |\Sigma|^{L-1})$ [29]. Thus, we can prune the *wPST* by using Pr_{min} or *MinCt*, which only increases the number of nodes exponentially at first a few levels and then decreases and converges to some constant *C*. However, using the Pr_{min} or *MinCt* to perform the pruning operation during the *wPST* construction process may result in the loss of the anomalous subsequence. In order to solve the above problem, this paper proposes a strategy to put the sequence corresponding to the node whose occurrence number is less than *MinCt* or occurrence probability is less than Pr_{min} into the candidate pattern anomalies set, and then analyzes and mines the candidate set to obtain pattern anomalies that meets the user's requirements.

During the *wPST* construction process, each node of the *wPST* stores the occurrence number of the string traversing from the root to this node, the occurrence probability of the node and the probability vector of the subsequent nodes. Hence, it only needs to analyze the node to determine if the sequence is a pattern anomaly or not during the *wPST* tree construction process; that is, if the sequence whose occurrence number is less than *MinCt* or the occurrence probability is less than Pr_{min}, then it puts the node corresponding to the sequence and all its descendant nodes into the candidate pattern anomalies set. The formal description of the candidate anomaly mining algorithm *Mine_Candidate_ Anomaly* is shown in Algorithm 2.

Algorithm 2 Candidate anomaly pattern mining. *Mine_Candidate_Anomaly* (*wPST T*, *int MinCt*, *real* Pr_{min})

Input: *wPST T, MinCt,* Pr_{min}
Output: candidate pattern anomaly set *cpas*
1. **Initialize:** *cpas*←∅
2. ***ForEach*** represent(*u*,*X*) ∈*T*
3. occ_times(*u*).Cal(); *Pr* (suffix(*u*)). Cal ();
4. If (occ_times(*u*) < *MinCt* || *Pr* (*u*) < Pr_{min})
5. *caps*.Add(represent(*u*,*X*));
6. *caps*.Add(descendants (represent(*u*,*X*)));
7. *T*.Prune(represent(*u*,*X*));
8. *T*.Prune(descendants (represent(*u*,*X*)));
9. ***Return*** *caps*

3.2.4. Pattern Anomalies Verification

Generally speaking, pattern anomalies have a higher probability coming from the candidate pattern anomalies set *caps*. However, there may be some special pattern anomalies that are not in the *cpas*; in addition, the *cpas* may also have partially redundant pattern anomalies. Hence, it is necessary to mine and analyze the *cpas* to obtain the pattern anomalies. The pattern anomalies mining mainly include:

(1) Pattern filtering: for pattern s_1 corresponding to node *u* and pattern s_2 corresponding to node *v* in the *cpas*, if pattern s_2 is a substring of the pattern s_1, add pattern s_1 to the pattern anomalies set *pas*.

(2) Pattern merging: for pattern s_1 corresponding to node *u* and pattern s_2 corresponding to node *v* in the *cpas*, if pattern s_1 and pattern s_2 have the longest common substring s_3; furthermore, s_3 is the true suffix of pattern s_1 and pattern s_2, then merge pattern s_1; (s_2-s_3) becomes the new pattern s' and is added to the pattern anomalies set *pas*; else add s_3 to *pas*, where '-' in (s_2-s_3) means deleting pattern s_3 from s_2.

(3) Pattern expanding: for each pattern $\sigma_i s_i$ corresponding to node u_i ($1 \le i \le |\Sigma|$) and its parents node *u* in *wPST*, if pattern $s\sigma$ corresponding to node *u* does not include in *caps* but all $\sigma_i s_i$ is included in *caps*, prune the parent node *u* corresponding to pattern $s\sigma$ from *wPST* and add $s\sigma$ to the pattern anomalies set *pas*.

(4) Pattern verifying: for each pattern $s\sigma$ in *cpas*, if there exists an alphabet $\sigma' \in \Sigma$, make the occurrence number of $s\sigma$ be equal to the occurrence number of $s\sigma\sigma'$; that is, the probability that the event σ' occurs after the event $s\sigma$ is 1. Although the probability of event $s\sigma$ is lower than *MinCt*, the occurrence of sσ represents the occurrence of a high confidence event $s\sigma\sigma'$. Therefore, $s\sigma$ cannot be simply treated as a pattern anomaly and should be deeply verified and analyzed.

(5) Pattern sorting: the probability of sequences corresponding to nodes in different levels of *wPST* is different. Generally, if the symbol sequence has the same occurrence number, the closer a node is to the root, the higher the probability it is to be an anomalous pattern. Thus, for pattern s_1 corresponding to node u_1 and pattern s_2 corresponding to node u_2 in the *pas*, if the occurrence number of s_1 equals the occurrence number of s_2 and the node u_1 is closer to root than u_2, it seems that s_1 has a higher probability to be an anomalous pattern than s_2. Therefore, the *top-k* anomalous patterns can be gained by using this rule to sort the patterns in *pas*.

The formal description of the pattern anomalies mining process is shown in Algorithm 3.

Algorithm 3 Anomalies Pattern Mining. *Mine_Anomaly* (*CAPS caps*)

Input: candidate pattern anomaly set *caps*
Output: pattern anomaly set *aps*
1. **Initialize:** *aps*←∅
2. Pattern_Filter*(caps);*
3. Pattern_Merge*(caps);*
4. Pattern_Extend*(caps);*
5. Pattern_Valid *(aps);*
6. Pattern_Sort*(aps);*
7. **Return** *aps*

3.3. Algorithm Analysis

TFSAX_wPST can be divided into four parts: time series symbolization *TFSAX*, *wPST* construction, candidate pattern anomalies generation and pattern anomalies verification. For *TFSAX*, it has been proven to have a slightly more time complexity than *SAX*, but can achieve better symbolization. For the second part, it prunes the *wPST* by using Pr_{min} or *MinCt*, thus the number of nodes only increases exponentially at first a few levels and then decreases and converges to some constant *C* [29]. Therefore, the total cost of constructing the *wPST* is approximately equal to $O(NmL) + O(L \times |\Sigma|^{\alpha}) + O(LC)$, where *N* is the total length of *S*, m is the average length of the sequence of *S*, $\alpha$is a fixed integer, which depends upon the pruning parameters (usually less than 4), and *C* is a constant. Since the probability of pattern anomalies is small, the number of nodes included in the candidate pattern anomalies set is far less than $|\Sigma|^{L-1}$. Therefore, the time complexity required for candidate pattern anomalies generation and pattern anomalies mining will be much lower than that of *wPST* construction. Hence, the time complexity of *TFSAX_wPST* is mainly concentrated on *TFSAX* representation and *wPST* construction. Theoretically, the performance and efficiency of our algorithm are effectively improved compared to *PST*-based methods.

4. Case Studies

In this section, we conduct a set of experiments to show the accuracy and feasibility of our new approach. Here we choose different datasets (the *NWIS* dataset and *Poyang Lake* dataset) for experiments to prove the generality of our model.

4.1. NWIS Dataset

4.1.1. Research Area

Echeconnee Creek (site number 02214075, location at 32°41′30.76″ N, 83°42′03.5″ E) is an important water level and flow control station in Peach County, Georgia. It is 9.1 miles from the confluence with

the Ocmulgee River, 4.4 miles northwest of Byron, GA, and its basin has an area of 228 square miles (Figure 4). This station is a typical hydrological station in the southern United States. Every year from July to October, the water level and discharge gradually decrease due to the influence of the Atlantic monsoon and will fall to the lowest value in September or October. With the increase of precipitation from November to June of the following year, the water level and discharge begin to rise and will reach the highest level from January to February. According to historical data, its monthly mean water level varies from 6.4 feet (September) to 9.3 feet (February) and its monthly mean discharge varies from 76 ft^3/s (October) to 452 ft^3/s (February).

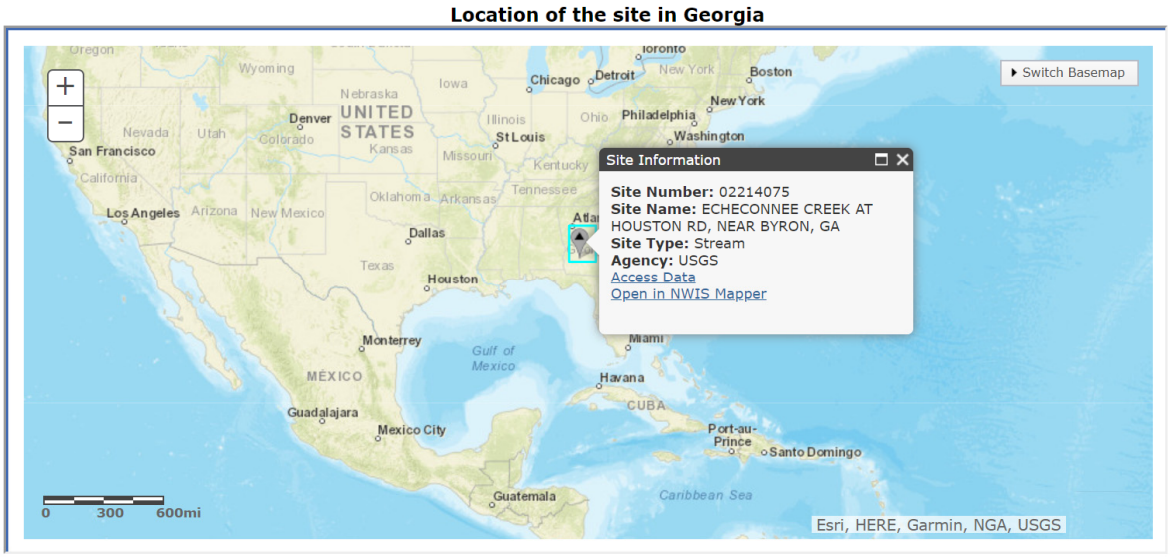

Figure 4. Echeconnee Creek station and its location in Georgia, USA.

Therefore, we used hourly water level, discharge and rainfall from 15 November 2010 to 15 November 2013 provided by the NWIS, USGS (NWIS: https://waterdata.usgs.gov/nwis/inventory/?site_no=02214075&agency_cd=USGS&amp), to verify the feasibility and effectiveness of this algorithm. The original water level data is shown in Figure 5. It should be mentioned that we used data quality control methods in the literature [15] and the point outlier detection method in the literature [13] to perform data quality control and point outlier detection on the original data set, so as to provide high-quality data for pattern anomalies detection.

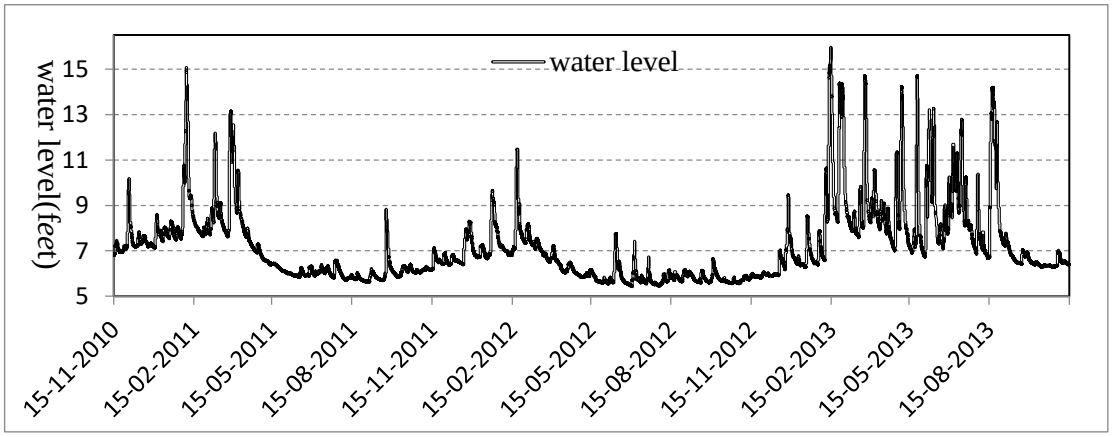

Figure 5. Real-time water level data of Echeconnee Creek.

4.1.2. *TFSAX* Representation

As can be seen from Figure 5, the water level data of Station 02214075 is smooth overall, but there are still some local "extreme" patterns that are obviously inconsistent with other patterns. In order to discover those "interesting" information in the series, we first use *TFSAX* to transform hydrological time series into symbolic sequence representation. In this experiment, we discretize the daily monitoring data (including 24 monitoring records with an interval of 1 h) into a mean symbol and a trend feature symbol representation. Therefore, the experimental time series will be divided into 1096 sequence segments (the total number of days from 15 November 2010 to 15 November 2013); that is, $w = 1096$. According to the *TFSAX*, the mean and trend feature of the given time series from 15 November 2010 to 15 November 2013 can be represented by 5 and 7 symbols. That means the number of mean symbols is $\alpha = 5$, and the number of trend feature symbols is $\alpha' = 7$. The character set represented by the mean and trend feature and their corresponding physical meanings are shown in Tables 1 and 2.

Table 1. The character sets and their meanings for the mean feature.

Symbol	Meaning (Water Level)
A	5.46 ft–7.45 ft
B	7.46 ft–9.43 ft
C	9.46 ft–10.39 ft
D	10.48 ft–13.44 ft
E	13.48 ft–16.56 ft

Table 2. The character sets and their meanings for the trend feature.

Symbol	Trend Feature	Meaning
a	(−90°–−45°)	water level drops sharply
b	(−90°–−30°)	water level drops rapidly
c	(−30°–−5°)	water level drops slowly
d	(−5°–5°)	water level remains stable
e	(5°–30°)	water level rises slowly
f	(30°–45°)	water level rises rapidly
g	(45°–90°)	water level rises sharply

After *TFSAX* symbolic representation, the original water level time series containing 1096×24 records will be symbolized into a symbol sequence containing 1096 symbols. The *TFSAX* representation of the water level time series is shown in Table 3.

Table 3. The *TFSAX* representation of the daily mean water level of station 02214075.

Symbol	Frequency	Symbol	Frequency	Symbol	Frequency	Symbol	Frequency	Symbol	Frequency
E_b	1	C_c	3	E_a	5	B_f	11	B_d	33
E_e	1	C_f	3	D_c	5	A_f	12	B_e	37
E_c	1	E_g	3	D_f	6	B_b	13	A_e	103
A_b	2	A_g	4	B_a	6	D_g	13	B_c	125
E_f	2	D_b	4	C_a	9	D_a	7	A_c	149
D_e	3	C_b	4	C_g	9	B_g	8	A_d	495

4.1.3. *wPST* Construction

As shown in Table 3, we can find some pattern anomalies. For example, pattern E_b means the water level is in state E (high water level between 13.48 and 16.56 feet) and the trend feature is in state b (the water level drops rapidly, and the trend feature angle is −45°–30°) is a rare pattern in the time series. It will be added to the candidate pattern anomaly set according to *TFSAX_wPST*. In order to analyze the symbolized sequence, we used the *wPST* construction algorithm *Build_wPST* to construct the *wPST* for the sequences shown in Table 3. For the convenience of description, it uses A_d with the constraint of the depth of tree $L \leq 3$ to illustrate the construction of *wPST*. The constructed *wPST* is shown in Figure 6.

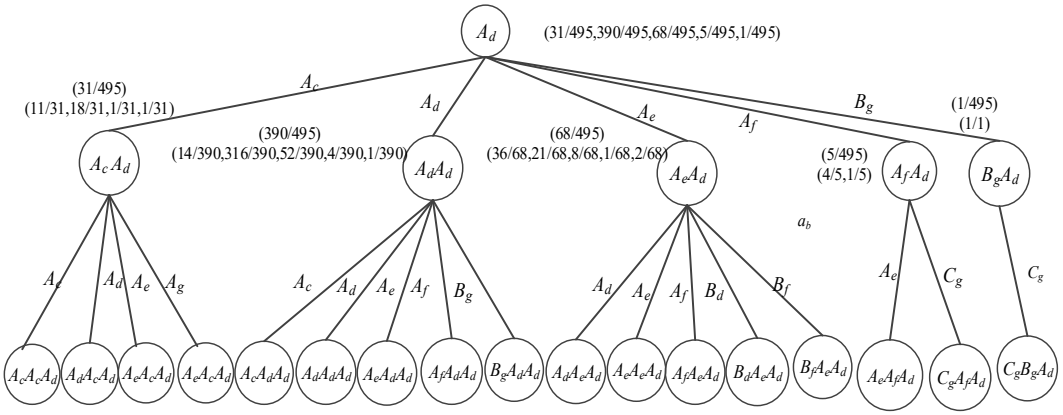

Figure 6. *wPST* tree representations for event A_d.

4.1.4. Detection Results and Analysis

From Figure 6, it can be inferred that the normal subsequent stages of state A_d should be A_c, A_d and A_e. Hence, it may indicate an anomalous event occurred if state A_f or B_g appears right after state A_d. Here we use the algorithm *Mine_Candidate_Anomaly and Mine_Anomaly* to detect those patterns that meet the anomaly pattern definition in Definition (2).

In this experiment, we set parameters $Pr_{min} = 0.01$ and $MinCt = 5$. When *wPST* is constructed, any node whose occurrence probability is less than Pr_{min} or occurrence number is less than *MinCt* will be pruned from *wPST.* Moreover, the sequences corresponding to those nodes and all of its descendant nodes will be put into the candidate pattern anomalies set *caps*. For example, the node A_fA_d and all its descendant nodes will be pruned from the *wPST* shown in Figure 6, and all the sequences that contain patterns A_dA_f (e.g., $A_dA_dA_f$) will be put into the *caps.*

After *caps* is generated, we will validate and analyze the patterns in it to determine the final pattern anomalies. Take A_dB_g for instance: we checked and analyzed the original data shown in Figure 5 and find that the pattern $A_dB_gC_g$ corresponds to the anomalous rain event from 15 August 2013 to 17 August 2013 in the Echeconnee Creek basin. On August 15, 16 and 17, the precipitation of this station was 1.41 in, 0.98 in and 1.45 in, respectively. As a result, the water level sharply rose 2.22 ft, 2.79 ft, 1.27 ft and 1.14 ft on 15 August–18 August, and the water level state represented by *TFSAX* changed drastically from A_d to B_g and then to C_g. Our method can quickly and accurately detect the pattern corresponding to this time series as an anomalous pattern. Similarly, the algorithm can also detect pattern anomalies, such as A_cA_e, A_cA_g, A_eB_d and A_fB_g, in a given time series.

The pattern anomalies detected by *TFSAX_wPST* on the Echeconnee Creek water level time series data set are shown in Table 4. The analysis of the results and corresponding events verifies that our method can effectively detect anomalous patterns, and thus provides high-quality data and knowledge support for subsequent hydrological analysis and application.

Table 4. Anomalous patterns detected by the algorithm and event descriptions.

Pattern	Subsequence	Corresponding Event Description
$B_gC_cB_a$	1 Dec 2010–3 Dec 2010	Daily water level is +1.88, −0.3, −1.19 feet, respectively.
$B_gD_eD_fD_gE_fE_aD_aC_b$	2 Feb 2011–9 Feb 2011	Daily water level is +1.68, +0.24, +1.33/2, +1.53, + 1.39/2, −1.53, −2.42, −0.8 feet, respectively.
$B_fC_gD_eC_aB_c$	9 Mar 2011–12 Mar 2011	Daily water level is +2.13, +0.36, −2.35, −0.53 feet, respectively.
$C_gD_gD_bD_cD_fD_bD_aB_bB_cB_gC_fC_a$	27 Mar 2011–7 Apr 2011	Daily water level is +3.38, +1.6, −0.86, −0.31, + 0.9, −0.65, −1.74, −0.7, −0.39, +1.09, +0.61, −1.25 feet, respectively.
$B_gB_cB_b$	23 Sep 2011–25 Sep 2011	Daily water level is +2.34/2, −0.51, −0.72 feet, respectively.
A_gB_g	21 Jan 2012–22 Jan 2012 2012.1.21–1.22	Daily water level is +1.3, +1.1 feet, respectively.
$A_fC_gD_fD_aB_cB_b$	18 Feb 2012–23 Feb 2012	Daily water level is +0.67, +2.67, +0.79, −2.1, −0.99 feet, respectively.
$B_fB_gB_aB_c$	26 Dec 2012–28 Dec 2012	Daily water level is +0.65, +1.49, −1.55 feet, respectively.
$A_gC_gC_aB_aC_gD_gE_eE_cE_aD_aC_aB_c$	7 Feb 2013–18 Feb 2013	Daily water level is +1.87, +1.78, −1.06, −1.14, +3.19, +3.44, +0.26, −0.32, −1.59, −2.04, −1.47, −0.57 feet, respectively.
$B_gD_gE_gE_aD_cD_cE_fE_aD_aC_aB_c$	22 Feb 2013–2 Mar 2013	Daily water level is +1.14, +2.86, +1.85, −1.14, −0.2, +1.31, −1.24, −2.1, −1.15, −0.46 feet, respectively.
$C_gD_gE_aD_aC_b$	24 Mar 2013–28Mar 2013	Daily water level is +3.25, 3.18/2, −1.62, −2.66, −0.88 feet, respectively.
$B_gD_gD_aB_bB_c$ $B_gD_gE_gD_aD_aB_b$	29 Apr 2013–9 May 2013	Daily water level is +2.04, +1.91, −1.98, −0.88, −0.31, +1.05, +2.55, +2.09/2, −1.31, −2.68, −0.9 feet, respectively.
$C_gE_gD_aB_a$	23 May 2013–26 May 2013	Daily water level is +4.21, +2.1/2, −3.97, −1.41 feet, respectively.
$B_gD_cB_fD_gD_fD_bD_aB_cD_gD_fD_a$	3 Jun 2013–13 Jun 2013	Daily water level is +3.03, −0.24, +0.78, +2.17, +0.57, −0.8, −1.7, 0.35, +2.42, +0.6, −3.46 feet, respectively.
$B_fC_fD_aD_eC_aD_gD_aB_a$	3 Jul 2013–10 Jul 2013	Daily water level is +0.52, +0.46, −1.25, +0.25, −1.2, +1.61, −1.16, −1.13 feet, respectively.
$C_gD_gD_fD_aB_a$	12 Jul 2013–16 Jul 2013	Daily water level is +1.97, +1.2, +0.57, −2.76, −1.11 feet, respectively.
$A_gB_gC_aB_b$	2013.7.31–8.3	Daily water level is +1.01, +2.43, −1.52, −0.92 feet, respectively.
$B_gC_gD_gD_gE_b$	15 Aug 2013–19Aug 2013	Daily water level is +2.22, +2.79, +1.27, +1.14 feet, respectively.

4.2. Poyang Lake Data Set

4.2.1. Research Area

Poyang Lake (Figure 7), the largest freshwater lake in China, is an important reservoir lake and an important international wetland in the mainstream of the Yangtze River. It is located on the south

bank of the middle reaches of the Yangtze River and north of Jiangxi Province. The catchment has a subtropical wet climate characterized by an annual mean precipitation of 1680 mm and an annual mean evaporation of 1200 mm. Poyang Lake receives water flows from five rivers: Ganjiang, Fuhe, Xinjiang, Raohe and Xiushui, and exchanges water with the Yangtze River. Lake storage and lake level variation is controlled by catchment discharges and interactions with the Yangtze River [44]. From April to June each year, the lake experiences large water level fluctuations in response to the catchment's annual cycle of precipitation. From July to September, it is affected by the backflushing or backwatering of the Yangtze River to maintain high water levels. In the wet season (April to September), the water level rises and the lake coverage expands, covering an area of roughly 170 km from the north to the south and 17 km from the east to the west. The lake shrinks to little more than a river during the dry season (October to March), exposing extensive floodplains and wetland areas that support migrating waterfowls and a variety of invertebrate species [45].

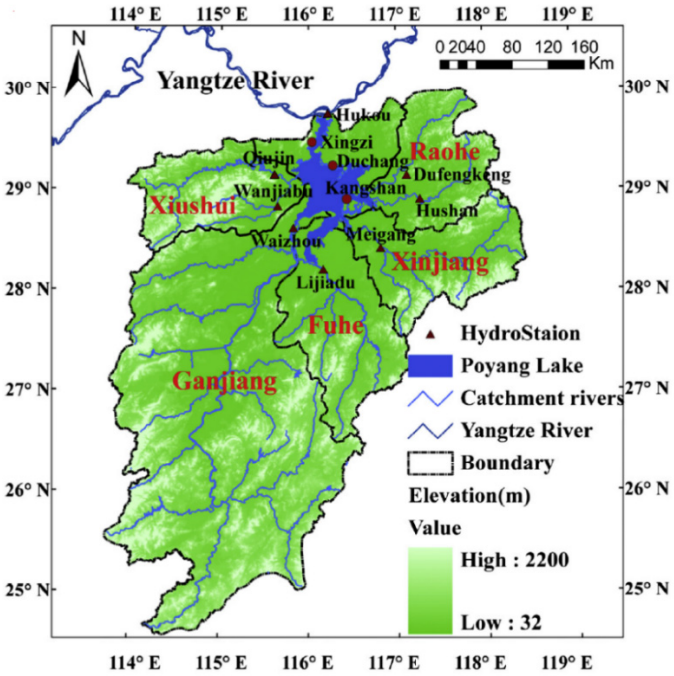

Figure 7. The Xingzi station and its location in the Poyang Lake and Yangtze River.

According to historical data, the multi-annual mean water level of Poyang Lake is 14.01 m; the monthly mean water level is highest in July (17.59 m) and lowest in January (10.52 m); and the highest water level appeared on 31 July 1998 (22.59 m), and the lowest appeared on 6 February 1963 (5.90 m). The Xingzi gauging station is the representative hydrological station of Poyang Lake and situated in the northern arm of the Lake at about 39 km from the Yangtze River. Typically, when the water level of Xingzi is below 11 m, it means that Poyang Lake has entered the dry season. Meanwhile, if the water level of Xingzi Station is above 19 m, it indicates that the water level of the Poyang Lake exceeds the warning line and is entering the flood season. The monthly mean lake water level at the Xingzi station from 1953 to 2009 is shown in Figure 8. We also used data quality control methods from the literature [15] and the point outlier detection method from the literature [13] to perform data quality control and point outlier detection on the original data set, so as to provide high-quality data for pattern anomalies detection.

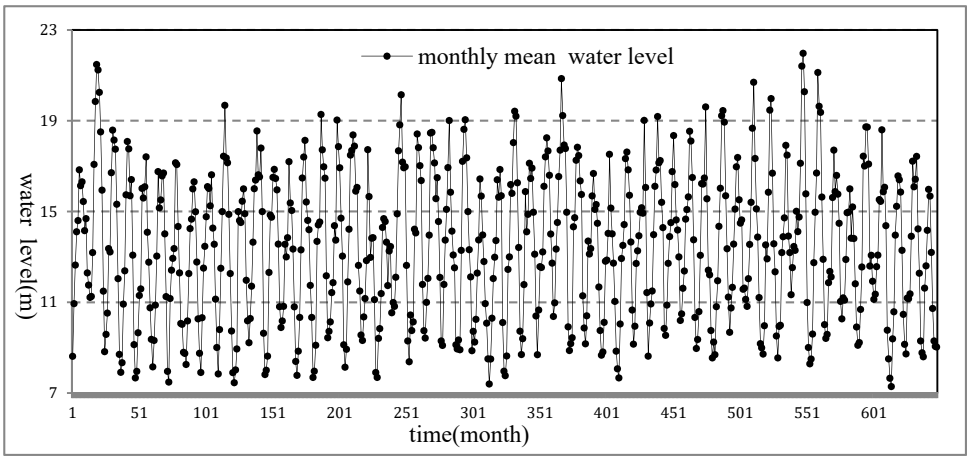

Figure 8. Monthly mean water level at the Xingzi Station.

4.2.2. *TFSAX* Representation

The monthly mean water level data of Xingzi Station is smooth overall in Figure 8, but there are still some local "extreme" patterns that are obviously inconsistent with other patterns. In order to discover those "interesting" patterns in this series, we first use *TFSAX* to transform the monthly mean water level data into a symbolic sequence representation.

In this experiment, we discretize the monthly statistics data (including 30 or 31 records with an interval of 1 day) into a mean symbol and a trend feature symbol representation. Therefore, the experimental time series will be divided into 684 sequence segments (the total number of month from January 1953 to December 2009); that is, $w = 684$. According to the *TFSAX*, both the mean and trend feature of the given time series from January 1953 to December 2009 would be represented by 5 symbols. That means the number of mean symbols is $\alpha = 5$, and the number of trend feature symbols is $\alpha' = 5$. The character set represented by the mean and trend feature and their corresponding physical meanings are shown in Tables 5 and 6.

Table 5. The character sets and their meanings for the mean feature.

Symbol	Meaning (Water Level)
A	7.28 m–8 m
B	8.01 m–10.99 m
C	11.03 m–15 m
D	15.04 m–19 m
E	19.01 m–21.96 m

Table 6. The character sets and their meanings for the trend feature.

Symbol	Trend Feature	Meaning
a	(−90°–−30°)	water level drops rapidly
b	(−30°–−5°)	water level drops slowly
c	(−5°–5°)	water level remains stable
d	(5°–30°)	water level rises slowly
e	(30°–90°)	water level rises rapidly

After the *TFSAX* symbolic representation, the original water level time series containing 684×30 records will be symbolized into a symbol sequence containing 684 symbols. The *TFSAX* representation of the water level time series is shown in Table 7.

Table 7. The *TFSAX* representation of the monthly average water level of the Xingzi station.

Symbol	Frequency	Symbol	Frequency	Symbol	Frequency	Symbol	Frequency
A_a	12	B_c	5	C_d	28	D_e	82
A_b	9	B_d	31	C_e	101	E_a	4
A_c	1	B_e	37	D_a	50	E_b	3
A_d	4	C_a	89	D_b	31	E_d	4
B_a	73	C_b	24	D_c	9	E_e	16
B_b	29	C_c	12	D_d	30		

4.2.3. *wPST* Construction

We can find some pattern anomalies in Table 7. For example, pattern B_a means when the water level is in state *B* (dry season, water level is 8–11 m), the trend feature is in state *a* (the water level drops rapidly, and the trend feature angle is −90°–−30°). It will be a rare pattern in the time series if the subsequent state *of* B_a is *C* (normal, water level is between 11 and 15 m) and the subsequent trend feature of B_a is *e* (water level rises rapidly, the trend feature angle is 30°–90°). It will be added to the candidate pattern anomaly set according to *TFSAX_wPST*.

In order to analyze the symbolized sequence, we used the algorithm *Build_wPST* to construct the *wPST* for the sequences shown in Table 7. For the convenience of description, it uses B_d under the constraint that the depth of tree $L \leq 3$ to illustrate the construction of *wPST*. The constructed *wPST* is shown in Figure 9.

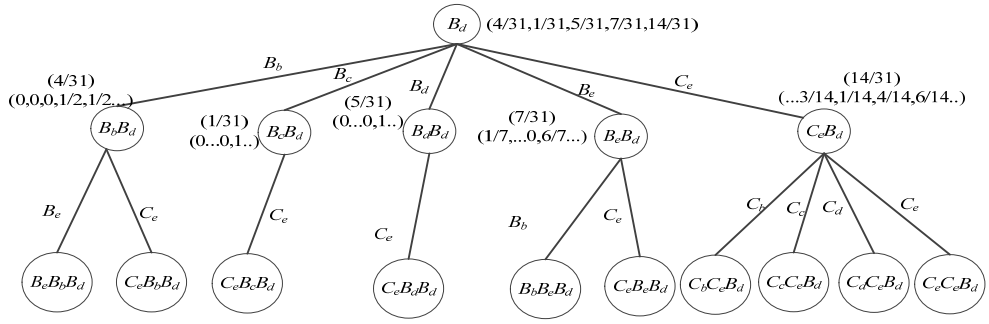

Figure 9. *wPST* tree representations for event B_d.

4.2.4. Detection Results and Analysis

From Figure 9, it can be inferred that state B_d (dry season, water level rises slowly) means the water level of Poyang Lake starts to rise slowly and its subsequent patterns is most likely to be B_d, B_e and C_e. So, it may indicate that an anomalous event occurred if states B_b or B_c appears right after state B_d. In order to detect those patterns that meet the anomaly pattern definition in Definition (2), we set parameters $Pr_{min} = 0.02$ and $MinCt = 4$.

When *wPST* is constructed, any node whose occurrence probability is less than Pr_{min} or occurrence number is less than *MinCt* will be pruned from *wPST*. Moreover, the sequences corresponding to those nodes and all of its descendant nodes will be put into the candidate pattern anomalies set *caps*. For example, the node B_cB_d and all its descendant nodes will be pruned from the *wPST* shown in Figure 9. Meanwhile, all the sequences that contain pattern B_dB_c (e.g., $B_dB_cB_e$) will be put into the *caps*.

After *caps* is generated, we will validate and analyze the patterns in it to determine the final pattern anomalies. Take pattern $B_dC_eC_c$ for instance: we checked and analyzed the original data shown in Figure 8. It shows that the pattern $B_dC_eC_c$ corresponds to the flood event from April to August 1974 in the Xingzi water level time series. Due to the influence of upper stream inflow from Ganjiang, Fuhe, Xinjiang, Raohe and Xiushui during the rainy season, the monthly mean water level of Xingzi Station

soared from 10.13 m in April 1974 to 14.21 m in May, and dropped slightly in June to 14.05 m; then, in July, it rose to 18.41 m (the highest water level is 20.1 m). Our method can quickly and accurately detect the pattern corresponding to this time series as an anomalous pattern. Similarly, our algorithm can also detect other pattern anomalies, such as $B_dB_eB_b$ corresponding to drought events at Poyang Lake from September 2006 to May 2007, and $B_dB_cC_e$ corresponding to drought events at Poyang Lake from December 2007 to January 2008.

The pattern anomaly results detected by *TFSAX_wPST* on the Xingzi water level time series data set are shown in Table 8. The analysis of the results and corresponding events verifies that our method can effectively detect anomalous patterns, and thus provides high-quality data and knowledge support for subsequent hydrological analysis and application.

Table 8. Anomalous patterns detected by the algorithm and event descriptions at the Xingzi station.

Pattern	Subsequence	Corresponding Event Description
$D_eE_eE_eE_bE_aD_aD_aC_a$	Mar 1954–Dec 1954	Extraordinary floods in the Yangtze River Basin, monthly mean water levels are 17.07, 19.84, 21.47, 21.23, 20.24, 18.5, 15.93, 11.48 m
A_aA_b	Dec 1958–Feb 1959	Extreme drought season, monthly mean water levels are 7.95, 7.48, 11.16 m
	Jan 1963–Oct 1963	Drought year, monthly mean water levels are 7.89, 7.45, 8.01, 8.94, 14.99, 14.59, 14.51, 15.45, 15.99, 14.9 m, highest water level occurred in September
	Dec 1971–Mar 1972	Extreme drought event, monthly mean water levels are 7.9, 7.68, 9.4, 9.83 m
	Dec 1979–Feb 1980	Extreme drought event, monthly mean water levels are 7.96, 7.76, 8.63, 12.4 m
A_aA_d	Jan 1965–Apr 1965	Extreme drought event, monthly mean water levels are 7.81, 8, 8.62, 12.31 m
	Jan 1968–Apr 1968	Extreme drought event, monthly mean water levels are 7.68, 7.9, 9.1, 13.68 m
	Dec 2007–Mar 2008	Extreme drought event, monthly mean water levels are 7.54, 7.72, 8.5, 8.62 m
$D_bD_eE_eE_bD_aC_a$	Jul 1980–Oct 1980	Flood events, monthly mean water levels are 18.03, 19.41, 19.19, 16.26 m
$E_eE_aD_aD_cC_a$	Jul 1983–Oct 1983	Flood events, monthly mean water levels are 20.85, 19.22, 17.9, 17.77 m
$C_cE_eD_a$	Jun 1968–Aug 1968	Flood events, monthly mean water levels are 14.53, 19.27, 17.71 m
$C_dC_bC_aC_aC_eC_dB_aB_d$	Jun 1972–Feb 1973	Drought year with a gentle overall trend, monthly mean water levels are 14.68, 14.55, 13.64, 11.73, 13.25, 13.45, 10.53, 10.98, 10.81 m
$B_aB_bA_bB_e$	Dec 1986–Feb 1987	Drought season, monthly mean water levels are 8.84, 8.06, 7.66, 8.34 m
$D_eE_eE_dE_aD_aB_a$	Jun 1998–Nov 1998	Extreme flood event, monthly mean water levels are 17.12, 21.4, 21.96, 20.17, 15.77, 10.98 m
$D_eE_eE_aE_bD_aC_a$	Jun 1999–Nov 1999	Flood events, monthly mean water levels are 16.69, 21.12, 19.63, 19.36, 15.63, 12.89 m
$C_aB_aB_aA_bA_bB_e$	Nov 2003–Mar 2004	Extreme drought event, monthly mean water levels are 8.5, 7.65, 7.28, 9.38 m
E_eE_d	Jul 1996–Oct 1996	Flood events monthly mean water levels are 19.46, 19.97 m
$B_dC_eC_cD_e$	Apr 1974–Jun 1974	No rainy season, monthly mean water levels are 10.13, 14.21, 14.05, 15.6 m
$B_aB_aB_bB_cB_aB_dB_eB_bB_e$	Sep 2006–May 2007	Extreme drought year, monthly mean water levels are 10.72, 9.29, 9.05, 9.02, 8.06, 8.28, 10.6, 10.4, 10.98 m

4.3. Analysis and Discussion

In order to verify the accuracy and efficiency of our method, we conducted three sets of comparative experiments. We first compared the construction efficiency and detection accuracy of the *wPST* and *PST* models. Then we compared the detection result of our algorithm with other different algorithms. Lastly, we compared the time complexity of our algorithm with other hydrological time series pattern anomaly detection algorithms. The following performance metrics—True Positive Rate (*TPR*), True Negative Rate (*TNR*), False Positive Rate (*FPR*), False Negative Rate (*FNR*), Accuracy, Precision, Recall and F1-score and Area Under the Curve (*AUC*) [46]—were used to evaluate the different approaches.

4.3.1. Construction Algorithm Comparison

Here, we first compared the construction efficiency between the *wPST* model and the traditional *PST* model on the Poyang Lake daily water level dataset. In this experiment, the performance is measured by the negative log-likelihood of the normal patterns given the observation of the anomalous patterns. Specifically, we constructed both *wPST* and *PST* models from the experimental data with Markov orders 1, 2, 3, 4 and 5. For each *wPST/PST* model, we calculate the negative log-likelihood $P(s|T)$ of the experiment sequence s based on the given *wPST/PST* model T. The larger the negative log-likelihood value is, the more dissimilar are the compared sequences. We expect the dissimilarity between the anomalous patterns and the normal patterns to grow as the memory order grows.

Our results are summarized in Table 9. The empirical results indicate that the sizes of the *wPST* model are much smaller than that of the *PST* model as the order increases. For example, the 5th order *PST* model uses 138 states to characterize the experimental dataset, while the 5th order *wPST* model only uses 84 states. The negative log-likelihood is the same between a sequence given a *wPST* model and a *PST* model with the same order, since we eliminate nodes that have the same probabilities as their parent nodes when constructing the *wPST* models. Therefore, we prefer a *wPST* model over a *PST* model because it is purely data-driven, flexible, and takes less space.

Table 9. Comparison of the *wPST* vs. *PST* model.

Approach	Order	Numbers of Nodes	−Log-Likelihood
PST-based	1	10	−0.0152
	2	46	−0.0108
	5	138	−0.0068
wPST-based	1	10	−0.0152
	2	41	−0.0108
	5	84	−0.0068

Note that the *PST* models can be pruned to remove some low probability nodes [29]; which will lead to information loss. Unlike *PST*, our approach prunes the low probability nodes and puts the sequence corresponding to those nodes and all its descendant nodes into the candidate pattern anomalies set *caps*, which can improve the accuracy and reduce the false detection rate of our algorithm.

Table 10 shows the confusion matrix obtained when adjusting the threshold *MinCt*. Based on the detection performances, the *FPR* for the *wPST* model on Poyang Lake dataset is 3.8% when $MinCt = 5$, which has the best tradeoff between *FPR* and *TPR*. As a comparison, the *FPR* for the *PST* model on Poyang Lake dataset is 21.9% when $MinCt = 5$, which has the best tradeoff between *FPR* and *TPR*. In addition, the miss rates for the *PST* and *wPST* models are 14.3% and 2.3%, respectively. The miss rates and false alarm rates are both relatively low for the *wPST* model. The detection results show that our proposed *TFSAX_wPST* algorithm is able to detect anomalies with a higher performance than that of the *PST* model.

Table 10. Performances for *wPST* and *PST*.

Approach	*MinCt*	FNR (Miss Rate)	FPR (False Alarm)
PST-Based (order = 5)	1	25.2%	64.9%
	2	22.7%	42.5%
	5	14.3%	21.9%
	10	12.5%	37.6%
wPST-Based (order = 5)	1	12.6%	25.6%
	2	8.4%	10.9%
	5	2.3%	3.8%
	10	5.3%	8.4%

4.3.2. Anomaly Detection Results Comparison

We compared the detection results of our algorithm with *PST*-based [41], *HMM*-based [31], *OCSVM* [47], *FP_SAX*-based [33], and *Distance*-based [34] algorithms on the same datasets. The detection results computed by the different algorithms on the Poyang Lake dataset are presented in Table 11. All results reported were averaged over 10 runs of both the representation learning and detection models.

Table 11. Anomaly detection results for the Poyang Lake dataset.

Metric \ Algorithm	*PST*-based	*HMM*-based	*OCSVM*	*FP_SAX*-based	*Distance*-based	*TFSAX_wPST*
Accuracy	0.912	0.928	0.874	0.936	0.947	0.976
Precision	0.926	0.922	0.896	0.927	0.935	0.964
Recall	0.925	0.932	0.902	0.944	0.951	0.969
F_1-score	0.926	0.927	0.918	0.935	0.943	0.966
AUC	0.924	0.931	0.915	0.938	0.949	0.971

The comparison results are displayed in the receiver operating characteristic (*ROC*) [48] curves shown in Figure 10. By convention, the *ROC* curve displays sensitivity (*TPR*) on the vertical axis against the complement of specificity (1 − specificity or *FPR*) on the horizontal axis. The *ROC* curve then demonstrates the characteristic reciprocal relationship between sensitivity and specificity, expressed as a tradeoff between the *TPR* and *FPR*. This configuration of the curve also facilitates calculation of the area beneath it as a summary index of the overall test performance. Therefore, the larger the area under the *ROC* curve, the better the performance of the technique is.

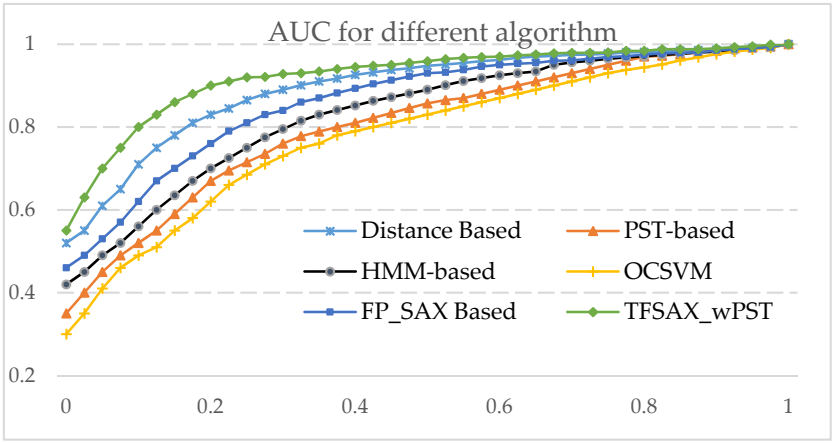

Figure 10. *AUC* for the different algorithms.

Figure 10 reveals the *AUC* obtained from the different algorithms. For experimental datasets, the *AUC* of the proposed algorithm are satisfactory and stable. These facts support the idea that our algorithm can effectively and accurately detect pattern anomalies and get better performances than that of *OCSVM*, *HMM*-based, *PST*-based, *FP*_SAX-based and *Distance*-based algorithms. This result is expected since the trend feature of the series is taken into account during the time series symbolization process *TFSAX*. Furthermore, we propose the improved probability suffix tree *wPST* to store the symbol sequence after *TFSAX* symbolization. Meanwhile, putting the symbol sequences pruned during the construction of *wPST* into *caps* rather than discarding them directly avoids information loss, which will improve the performance and efficiency of the algorithm.

4.3.3. Computational Complexity Comparison

One important aspect of anomaly detection is efficiency. In the hydrological field, it is important to ensure that the pattern anomalies are computed in a short amount of time and with a minimum delay. In this section, we compare the execution time of the *TFSAX_wPST*, *FP_SAX*-based [33] and *Distance*-based [34] algorithms along with the increase of the length of sequences on the Poyang Lake daily water level. The comparison results are shown in Table 12 and Figure 11.

Table 12. The execution time of different approaches.

Num	Time	Sequence Lengths	Total Length	*FP_SAX*-based	*Distance*-based	*TFSAX_wPST*
1	Jul–Aug	62	3534	0.372 s	0.324 s	0.264 s
2	Jun–Aug	92	5244	0.719 s	0.708 s	0.532 s
3	Jun–Sep	122	6954	1.133 s	1.012 s	0.793 s
4	Jun–Oct	153	8721	1.471 s	1.274 s	0.946 s
5	May–Oct	184	10,488	2.641 s	2.416 s	1.387 s
6	Apr–Nov	244	13,908	3.325 s	2.782 s	1.764 s

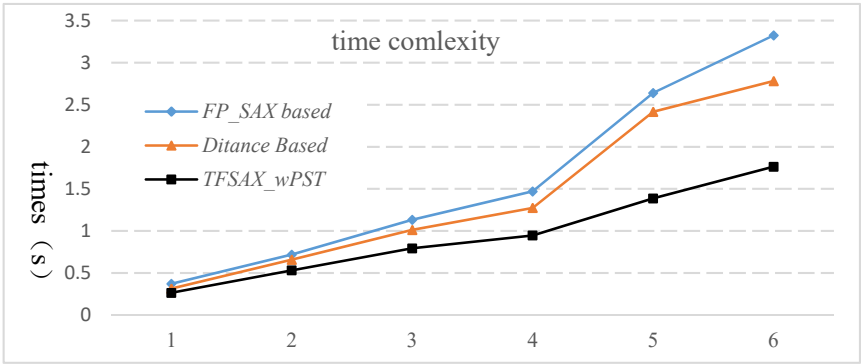

Figure 11. A comparison of the runtime of different approaches.

From Table 12 and Figure 11, we can see that the execution time of *TFSAX_wPST* is obviously less than that of the *FP_SAX*-based method [33] and *Distance*-based method [34]. The main reason is that the *FP_SAX*-based method and *Distance*-based methods need to measure distance between patterns and result in relatively high time complexity. As discussed in Section 3.3, the time complexity of our approach is mainly concentrated on *TFSAX* representation and *wPST* construction. The time of *TFSAX* symbolization is slightly better than that of *FP_SAX*, but our algorithm does not need to calculate the distance between patterns, so the time complexity is greatly improved.

5. Conclusions

In this paper we have conducted in-depth research on time series anomaly patterns and their detection algorithms; particularly, a detailed analysis of the framework, advantages and disadvantages, as well as an improvement strategy for the *wPST*-based approach. Combining with the field of hydrology, we proposed an effective and accurate anomaly pattern detection approach *TFSAX_wPST* for hydrological time series. At present, it mainly uses a distance-based approach to detect anomalous patterns in hydrological time series; however, the time complexity to calculate the distance between each pattern is very high. In this work, we combined symbolization (*TFSAX*) of time series with the VMM model (*wPST*). Then, a new approach that is suitable for hydrological time series anomalous pattern detection is put forward, which makes the detection results accurate and efficient.

There are some parts that remain to be improved in the future. Firstly, in the candidate pattern anomalies mining step, the threshold Pr_{min} or *MinCt* to prune the *wPST* is based on the experience of previous experiments. In the future we should consider a more scientific way of evaluation, which achieves the optimal value of Pr_{min} or *MinCt*. Secondly, compared to the fixed-length segmentation method *TFSAX*, how to use variable-length segmentation to represent time series for hydrological feature extraction is a more meaningful and interesting question. Finally, our approach mainly analyzes univariate time series anomalous pattern detection; therefore, how to apply this approach to detect multivariate hydrological time series anomalous patterns is a topic for future research.

Author Contributions: Conceptualization, Y.Y. and Q.Z.; data curation, Y.Y.; formal analysis, Y.Y. and H.L.; funding acquisition, D.W.; investigation, Y.Y.; methodology, Y.Y. and Q.Z.; project administration, D.W. and H.L.; software, Y.Y.; supervision, Y.Y. and Q.Z.; validation, D.W.; visualization, Q.Z.; writing—original draft, Y.Y.; writing—review and editing, D.W., Q.Z. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China grant number (No.2018 YFC1508100), the CSC Scholarship, and the Fundamental Research Funds for the Central Universities grant number (No.2018 B45614). And the APC was funded by (No.2018 YFC1508100).

Acknowledgments: This work is supported by the National Key Research and Development Program of China (No.2018 YFC1508100), the CSC Scholarship and the Fundamental Research Funds for the Central Universities (No.2018 B45614).

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

References

1. Chen, L.; Wang, L. Recent advance in earth observation big data for hydrology. *Big Earth Data* **2018**, *2*, 86–107. [CrossRef]
2. Guo, H.; Wang, L.; Chen, F.; Liang, D. Scientific big data and digital earth. *Chin. Sci. Bull.* **2014**, *59*, 5066–5073. [CrossRef]
3. Azimi, S.; Moghaddam, M.A.; Monfared, S.A. Anomaly Detection and Reliability Analysis of Groundwater by Crude Monte Carlo and Importance Sampling Approaches. *Water Resour. Manag.* **2018**, *32*, 4447–4467. [CrossRef]
4. Rougé, C.; Ge, Y.; Cai, X. Detecting gradual and abrupt changes in hydrological records. *Adv. Water Resour.* **2013**, *53*, 33–44. [CrossRef]
5. Hawkins, D.M. *Identification of Outliers*; Chapman and Hall: London, UK, 1980.
6. Chandala, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv. CSUR* **2009**, *41*, 1–58. [CrossRef]
7. Gupta, M.; Gao, J.; Aggarwal, C.; Han, J. Outlier detection for temporal data. *Synth. Lect. Data Min. Knowl. Discov.* **2014**, *5*, 1–129. [CrossRef]
8. USGS. Interagency Advisory Committee on Water Data. In *Guidelines for Determining Flood Flow Frequency: Bulletin 17 B*; U.S. Geological Survey, Office of Water Data Coordination: Reston, VA, USA, 1982.
9. Stedinger, J.R.; Griffis, V.W. Flood frequency analysis in the united states: Time to update. *J. Hydrol. Eng.* **2008**, *13*, 199–204. [CrossRef]

10. Chebana, F.; Daboniang, S.; Ouarda, T.B. Exploratory functional flood frequency analysis and outlier detection. *Water Resour. Res.* **2012**, *48*, 1–20. [CrossRef]
11. Sarraf, A.P. Flood outlier detection using PCA and effect of how to deal with them in regional flood frequency analysis via L-moment method. *Water Resour.* **2015**, *42*, 448–459. [CrossRef]
12. Amin, M.T.; Rizwan, M.; Alazba, A.A. Comparison of mixed distribution with EV1 and GEV components for analyzing hydrologic data containing outlier. *Environ. Earth Sci.* **2015**, *73*, 1369–1375. [CrossRef]
13. Yu, Y.; Zhu, Y.; Li, S.; Wan, D. Time series outlier detection based on sliding window prediction. *Math. Probl. Eng.* **2014**. [CrossRef]
14. Ng, W.W.; Panu, U.S.; Lennox, W.C. Chaos based analytical techniques for daily extreme hydrological observations. *J. Hydrol.* **2007**, *342*, 17–41. [CrossRef]
15. Zhao, Q.; Zhu, Y.; Wan, D.; Yu, Y.; Cheng, X. Research on the Data-Driven quality control method of hydrological time series data. *Water* **2018**, *10*, 1712. [CrossRef]
16. Nyeko-Ogiramoi, P.; Willems, P.; Ngirane-Katashaya, G. Trend and variability in observed hydrometeorological extremes in the Lake Victoria basin. *J. Hydrol.* **2013**, *489*, 56–73. [CrossRef]
17. Wang, C.; Zhao, Z.; Gong, L.; Zhu, L.; Liu, Z.; Cheng, X. A distributed anomaly detection system for in-vehicle network using HTM. *IEEE Access* **2018**, *6*, 9091–9098. [CrossRef]
18. Van Vlasselaer, V.; Bravo, C.; Caelen, O.; Eliassi-Rad, T.; Akoglu, L.; Snoeck, M.; Baesens, B. APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decis. Support Syst.* **2015**, *75*, 38–48. [CrossRef]
19. Golmohammadi, K.; Zaiane, O.R. Time series contextual anomaly detection for detecting market manipulation in stock market. In Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 19–21 October 2015; pp. 1–10.
20. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6479–6488.
21. Keogh, E.; Lin, J.; Fu, A. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In Proceedings of the IEEE International Conference on Data Mining, Houston, TX, USA, 27–30 November 2005; IEEE Computer Society: Washington, DC, USA, 2005; pp. 226–233.
22. Candelieri, A. Clustering and support vector regression for water demand forecasting and anomaly detection. *Water* **2017**, *9*, 224. [CrossRef]
23. Yu, Y.; Zhu, Y.; Wan, D.; Liu, H.; Zhao, Q. A Novel Symbolic Aggregate Approximation for Time Series. In Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication, IMCOM 2019, Phuket, Thailand, 4–6 January 2019; Springer: Cham, Switzerland, 2019; pp. 805–822.
24. Ding, Z.; Fei, M. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proc. Vol.* **2013**, *46*, 12–17. [CrossRef]
25. Budalakoti, S.; Srivastava, A.N.; Otey, M.E. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2009**, *39*, 101–113. [CrossRef]
26. Safin, A.M.; Burnaev, E. Conformal kernel expected similarity for anomaly detection in time-series data. *Adv. Syst. Sci. Appl.* **2017**, *17*, 22–33.
27. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection for discrete sequences: A survey. *IEEE Trans. Knowl. Data Eng.* **2010**, *24*, 823–839. [CrossRef]
28. Keogh, E.; Lonardi, S.; Chiu, B.Y. Finding surprising patterns in a time series database in linear time and space. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 550–556.
29. Sun, P.; Chawla, S.; Arunasalam, B. Mining for Outliers in Sequential Databases. In Proceedings of the SIAM International Conference on Data Mining, Bethesda, MD, USA, 20–22 April 2006; 2006; pp. 94–105.
30. Klerx, T.; Anderka, M.; Büning, H.K.; Priesterjahn, S. Model-based anomaly detection for discrete event systems. In Proceedings of the International Conference on Tools with Artificial Intelligence, Limassol, Cyprus, 10–12 November 2014; pp. 665–672.

31. Zohrevand, Z.; Glasser, U.; Shahir, H.Y.; Tayebi, M.A.; Costanzo, R. Hidden Markov based anomaly detection for water supply systems. In Proceedings of the International Conference on Big Data, Washington, DC, USA, 5–8 December 2016; pp. 1551–1560.
32. Pimentel MA, F.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* **2014**, *99*, 215–490. [CrossRef]
33. Wan, D.; Xiao, Y.; Zhang, P.; Feng, J.; Zhu, Y.; Liu, Q. Hydrological time series anomaly mining based on symbolization and distance measure. In Proceedings of the 2014 IEEE International Congress on Big Data, Beijing, China, 27 June–2 July 2014; pp. 339–346.
34. Zhang, P.; Leung, H.; Xiao, Y.; Feng, J.; Wan, D.; Li, W.; Leung, H. A New Symbolization and Distance Measure Based Anomaly Mining Approach for Hydrological Time Series. *Int. J. Web Serv. Res.* **2016**, *13*, 26–45. [CrossRef]
35. Wu, H.; Li, X.; Qian, H. Detection of Anomalies and Changes of Rainfall in theYellow River Basin, China, through Two Graphical Methods. *Water* **2018**, *10*, 15. [CrossRef]
36. Ye, N. A markov chain model of temporal behavior for anomaly detection. In Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, NY, USA, 6–7 June 2000; Volume 166, p. 169.
37. Ron, D.; Singer, Y.; Tishby, N. The power of amnesia: Learning probabilistic automata with variable memory length. *Mach. Learn.* **1996**, *25*, 117–149. [CrossRef]
38. Bejerano, G.; Yona, G. Variations on probabilistic suffix trees: Statistical modeling and prediction of protein families. *Bioinformatics* **2001**, *17*, 23–43. [CrossRef]
39. Yang, J.; Wang, W. CLUSEQ: Efficient and effective sequence clustering. In Proceedings of the 19th International Conference on Data Engineering, Bangalore, India, 5–8 March 2003; pp. 101–112.
40. Kholidy, H.A.; Yousof, A.M.; Erradi, A.; Abdelwahed, S.; Ali, H.A. A Finite Context Intrusion Prediction Model for Cloud Systems with a Probabilistic Suffix Tree. In Proceedings of the 2014 European Modelling Symposium, Pisa, Italy, 21–23 October 2014; pp. 526–531.
41. Li, Y.; Thomason, M.; Parker, L.E. Detecting time-related changes in Wireless Sensor Networks using symbol compression and Probabilistic Suffix Trees. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, China, 18–22 October 2010; pp. 2946–2951.
42. Farahani, I.V.; Chien, A.; King, R.E.; Kay, M.G.; Klenz, B. Time Series Anomaly Detection from a Markov Chain Perspective. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1000–1007.
43. Keogh, E.; Chakrabarti, K.; Pazzani, M.; Mehrotra, S. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowl. Inf. Syst.* **2000**, *3*, 263–286. [CrossRef]
44. Hu, Q.; Feng, S.; Guo, H.; Chen, G.; Jiang, T. Interactions of the Yangtze river flow and hydrologic processes of the Poyang Lake, China. *J. Hydrol.* **2007**, *347*, 90–100. [CrossRef]
45. Li, X.; Zhang, Q.; Ye, X. Dry/wet conditions monitoring based on TRMM rainfall data and its reliability validation over Poyang Lake Basin, China. *Water* **2013**, *5*, 1848–1864. [CrossRef]
46. Han, J.; Jian, P.; Micheline, K. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2011.
47. Ghafoori, Z.; Erfani, S.M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C.A. Anomaly Detection in Non-stationary Data: Ensemble based Self-Adaptive OCSVM. In Proceedings of the International Joint Conference on Neural Networks, Vancouver, BC, Canada, 25–29 July 2016.
48. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]