sensors

MDPI

Article

A Study of Pattern Prediction in the Monitoring Data of Earthen Ruins with the Internet of Things

Yun Xiao [1], Xin Wang [1,2,*], Faezeh Eshragh [2], Xuanhong Wang [3], Xiaojiang Chen [1] and Dingyi Fang [1]

1 School of Information Science and Technology, Northwest University, Xi'an 710021, China; yxiao@nwu.edu.cn (Y.X.); xjchen@nwu.edu.cn (X.C.); dyfang@nwu.edu.cn (D.F.)
2 Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; feshrgh@ucalgary.ca
3 Department of Communication, Xi'an University of Posts & Telecommunications, Xi'an 710121, China; wxh@xupt.edu.cn
* Correspondence: xcwang@ucalgary.ca; Tel.: +1-403-220-3355

Academic Editor: Vittorio M. N. Passaro
Received: 3 March 2017; Accepted: 4 May 2017; Published: 11 May 2017

Abstract: An understanding of the changes of the rammed earth temperature of earthen ruins is important for protection of such ruins. To predict the rammed earth temperature pattern using the air temperature pattern of the monitoring data of earthen ruins, a pattern prediction method based on interesting pattern mining and correlation, called PPER, is proposed in this paper. PPER first finds the interesting patterns in the air temperature sequence and the rammed earth temperature sequence. To reduce the processing time, two pruning rules and a new data structure based on an R-tree are also proposed. Correlation rules between the air temperature patterns and the rammed earth temperature patterns are then mined. The correlation rules are merged into predictive rules for the rammed earth temperature pattern. Experiments were conducted to show the accuracy of the presented method and the power of the pruning rules. Moreover, the Ming Dynasty Great Wall dataset was used to examine the algorithm, and six predictive rules from the air temperature to rammed earth temperature based on the interesting patterns were obtained, with the average hit rate reaching 89.8%. The PPER and predictive rules will be useful for rammed earth temperature prediction in protection of earthen ruins.

Keywords: pattern prediction; multivariate sequential data; earthen ruin

1. Introduction

Earthen ruins are ancient ruins based on mud brick, rammed earth, or any type of construction using soil as the main building material and have great historical, cultural and scientific value [1]. Lacking the very architectural devices necessary for survival as originally designed (e.g., protective roofs and plastered surfaces), earthen ruins exist in a continual state of 'unbecoming' [2]. At the micro level, earth swelling and shrinkage coupled with decohesion of the earth-silt-sand agglomerate are the fundamental mechanisms responsible for soil destabilization, which will eventually lead to various scales of damage from the loss of surface and surface finishes. Most of these earthen ruins will either collapse over time from differential erosion, or eventually stabilize as unrecognizable lumps [2]. Figure 1 shows the cracking and weathering conditions (marked by circles) in the Ming Dynasty Great Wall in Shaanxi Province of China. Research shows that dramatic rammed earth temperature changes are one of the major reasons causing the destruction of earthen ruins [3,4]. Therefore an understanding of the changes of the rammed earth temperature of earthen ruins is important for protection of earthen ruins.

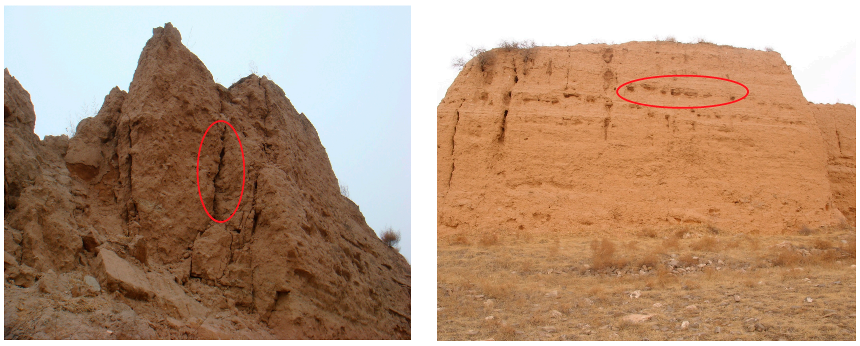

(**a**) cracking (**b**) weathering

Figure 1. Cracking and weathering damage in Ming Dynasty Great Wall.

The technology of the Internet of Things (IoT) has become an ideal solution for monitoring earthen ruins, because it is easily deployable and suitable for long-term and real-time data collection in remote areas. Most of the existing early studies have focused on collecting the monitoring data, but lack suitable data processing procedures [5–7]. Subsequent research has begun to focus on IoT monitoring data processing in the field of cultural heritage [8–12]. However, it is difficult for the earthen ruins conservation experts to make intelligent and scientific based decisions for protecting earthen ruins based on collected IoT data [13] due to the following challenges:

(1) It is difficult to describe the complex and interesting hidden relationship between internal earthen ruin parameters and external environmental parameters. For example, Figure 2 shows a plot of air temperature (line with $\bigcirc$) and rammed earth temperature of an earthen ruin (line with $\times$) sequences. As shown in the figure, in the area indicated by the left dashed box, both air temperature and rammed earth temperature values decrease. However, in the area indicated by the right dashed box, although the air temperature falls, the rammed earth temperature shows no significant change. In order to predict the rammed earth temperature trend based on the air temperature, we must know what kind of change in temperature is meaningful for earthen ruins conservation experts. However, not all the air temperature changes are interesting; for instance, earthen ruins conservation experts may only pay attention to air temperature changes that would cause damage in earthen ruins.

(2) With the increase in the number of IoT nodes, as well as the continuous monitoring, we will face dealing with huge amounts of time-series data. For example, the Ming Dynasty Great Wall ruins are more than 6000 km in length; if monitoring nodes were deployed every meter along the Ming Dynasty Great Wall, the number of nodes would be more than 6 million. Faced with large amounts of time-series data, we expect algorithms to provide prediction results efficiently.

(3) There is a hysteresis effect in rammed earth temperatures relative to air temperatures. Therefore, in the monitoring data of earthen ruins, a change in the rammed earth temperature has a certain delay relative to the change in air temperature. The delay parameter needs to be studied to be able to predict the exact rammed earth temperature trends with time.

To solve the aforementioned challenges, we propose a method called the Pattern Prediction on the monitoring data of Earthen Ruins (PPER) to acquire rammed earth temperatures of earthen ruins from IoT data. The proposed PPER can describe the interesting patterns of the earthen ruin variables by efficiently discovering the hidden relations between the time series of two correlated variables and pattern prediction. Specifically, the contributions of this paper are:

(1) Some terms for interesting patterns for earthen ruin monitoring are formalized. Terms, such as interesting pattern, direction, delay and variation of rule, are properly defined to precisely describe the patterns and changes in IoT data.

(2) The air temperature and rammed earth temperature are used as examples to show how the hidden relationships between the two earthen ruin monitoring data are discovered. Since dramatic temperature change is one major cause for the destruction of earthen ruins, we focus on the rising and falling of the temperature. PPER searches the whole dataset to find the rising and falling pattern and obtains the predictive rules to predict the rammed earth temperature.

(3) Two pruning rules are proposed to reduce the number of computations. At the same time, a new data structure version based on the R-tree [14] data structure is used to group similar patterns. Both techniques can significantly reduce the time complexity when processing huge amounts to IoT data.

(4) A set of experiments were conducted on the monitoring data of the Ming Dynasty Great Wall to demonstrate the effectiveness and efficiency of PPER. Interesting relationships between the air temperature and the rammed earth temperature were discovered to predict the rammed earth temperature pattern.

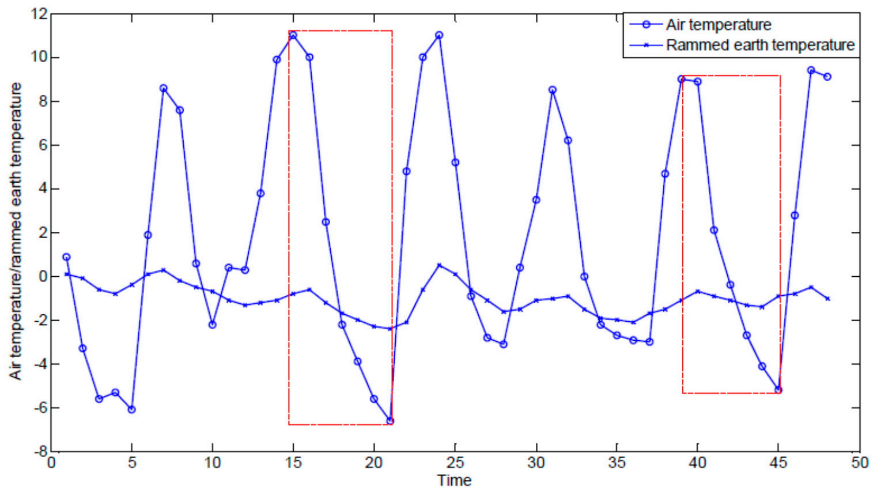


Figure 2. Air temperature and rammed earth temperature sequences.

The rest of this paper is organized as follows: Section 2 briefly reviews the most related works to this study. In Section 3, the required definitions are presented. In Section 4, the proposed method is described in detail. Section 5 presents the experimental studies and the results. Finally, the concluding remarks and some ideas for further studies are discussed in Section 6.

2. Related Work

In this section, we present and discuss related studies on: (i) destructive, nondestructive and micro-destructive measurements for acquiring rammed earth temperatures of earthen ruins; (ii) IoT technologies in the cultural heritage domain, and (iii) time-series prediction methods.

2.1. Destructive, Nondestructive and Micro-Destructive Measurements for Earthern Site Monitoring

In order to acquire rammed earth temperatures of earthen ruins, there are a few options, which can be groups into destructive, nondestructive and micro-destructive measurements. With destructive measurement methods, such as using a dynamic nuclear polarization (DNP) series of temperature probes and digital thermometers, the rammed earth temperature can be accurately obtained, but the earthen ruin will be damaged to a certain degree [15]. Nondestructive measurement methods, such as infrared imaging systems, can be used to get the surface temperature of rammed earth [15,16]. Using this measurement method, only the rammed earth surface temperature can be obtained, and the internal temperature of the rammed earth is still unknown. The third type is micro-destructive measurement methods, such as measurements obtained using the IoT.

Large amounts of monitoring data, such as the air temperature, can be obtained with IoT methods, such as the rammed earth temperature acquired by IoT nodes. A prediction model for IoT monitoring data based on a linear function between the air temperature and the rammed earth temperature was proposed to predict the rammed earth temperature of an earthen ruin in a completely closed environment [4]. Sufficient information on the rammed earth temperature with as little damage to the earthen ruin as possible was obtained. However, not all earthen ruins can be a completely closed environment, and the relation between the air temperature and the rammed earth temperature cannot always by satisfied with a linear function. For example, as shown in Figure 3, there is no simple linear relationship between the air temperature and the rammed earth temperature in the monitoring dataset of the Ming Dynasty Great Wall.

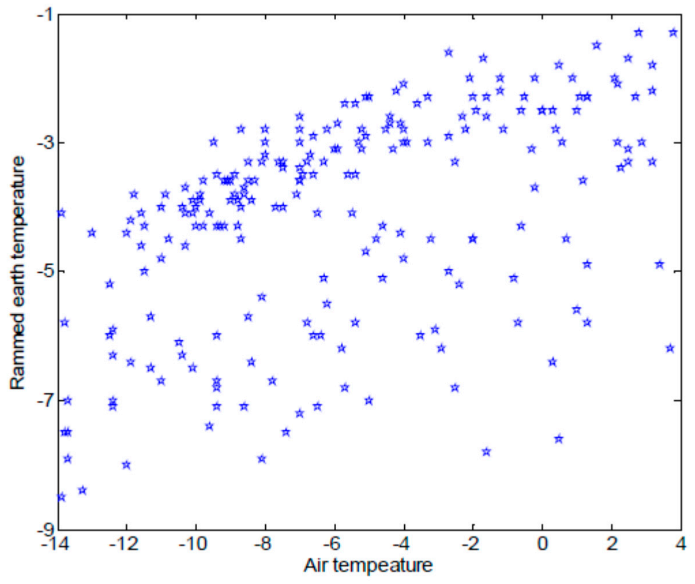


Figure 3. Relationship of the air temperature and the rammed earth temperature in Ming Dynasty Great Wall dataset.

2.2. *IoT Technology in Culture Heritage Domain*

IoT technologies have made great progress in cultural heritage applications. These technologies have two phases, the data monitoring and the data processing phase.

In the data monitoring phase, for example, Abrardo and Rodriguez-Sanchez used the Internet of Things technology collect environmental information of heritage sites [5]. In addition, the Institute of Computing Technology Chinese Academy of Sciences has developed an intelligent environment monitoring system which collected environmental information such as temperature, humidity, light at the Palace Museum heritage site [6]. A new sensor network for indoor environmental monitoring has been developed [7].

In the data monitoring and data processing phase, there have been some interesting applications in the field of cultural heritage based on IoT technology. In [8], an IoT architecture was designed to support the design of a smart museum based on an innovative model of sensors and services. An intelligent IoT system, designed with the aim of improving user experience and knowledge diffusion within a cultural space, was presented in [9]. Typical IoT smart technologies represent an effective mean to support users' understanding of cultural heritage [10]. An authoring platform, named FEDRO was presented automatically generate textual and users profiled artworks biographies, employed to feed a smart app for guiding visitors during the exhibition. An indoor location-aware architecture was designed and validated able to enhance the user experience in a museum [11]. In particular, the proposed system relies on a wearable device that combines image recognition and localization capabilities to automatically provide the users with cultural contents related to the observed artworks. A collaborative reputation system (CRS) was designed to establish the people reputation within cultural spaces [12].

2.3. Time-Series Prediction

Time-series prediction has many practical applications, such as weather forecasting and stock market prediction and therefore it has attracted a great deal of attention [17]. A large number of studies on time-series forecasting have utilized statistical models, such as autoregressive integrated moving average (ARIMA) [18] or exponential smoothing [19–21]. These models have been widely used for financial data analyses. However, the nonlinear and irregular nature of real-world time-series data has always been a problem. A newer approach to solving this problem uses machine learning techniques to predict the future results based on the knowledge learned from available data. For instance, artificial neural networks (ANN) have been widely used in time-series prediction [22–24]. However, disadvantages, such as over fitting and a time-consuming training phase, occur. Other techniques applied to solving this problem are support vector machines (SVM) [25] and the k-nearest-neighbor method [26].

Since many time-series variables are exposed to translation or dilatation in time, those approaches used traditionally for behavior forecasting will fail in providing useful hints about the future. A solution takes the behavior of the sequences into account rather than the exact values. For example, [27,28] proposed methodologies for predicting patterns in time series. Both of these studies suggested a new representation of data and then tried to find the most frequent patterns. However, these solutions had some problems. The data representation proposed in these solutions did not reduce the dimensionality of the data, especially for highly variant data. Therefore, a lot of data processing, like clustering, was necessary in [27], which can result in high time complexity.

Another problem was the inability to interpret the output rules and relationships. For instance, in [26] the relationship between patterns was defined using Allen's interval relations [29], which did not provide enough information for calculating delay. Reference [28] had the same problem for extracting information, such as sensitivity and direction of the variations, since it used symbolic aggregation approximation (SAX) representation, which summarized the data sequences based on the average value of intervals and did not pay attention to direction of variations.

These approaches try to learn from previously seen or predicted data and then predict the future values. However, if they use predicted values as an input for the next prediction step, error accumulation problem is possible [17]. Moreover, in reality, it is highly probable that the values of a time series are influenced by variations of other groups of time series.

Studies about multivariate forecasting methods take this problem into account and aim at solving it using statistical or artificial approaches. Neural networks [30–32] are one of the most important techniques applied in this area. The proposed methods try to find the future value of a time series based on previously seen data. However, in many real world situations, we are more interested in the upcoming trend is of more interest, not the exact and specific values. To achieve this goal, some studies have been done on univariate time-series data, using a set of previously observed values, called a pattern, to predict a set of future value(s) [33–36]. These methodologies try to find the hidden trends in the data and predict the future using the discovered patterns.

Predicting multivariate time-series data using patterns has been studied with interval data knowledge discovery [27,28,32]. Hoppner's work was aimed at solving a problem similar to the current study [27]. The author tried to find frequent patterns and use them for mining hidden rules in the data. However, their approach was different from our methodology in some aspects. They partitioned the sequences based on different data trends and coded the data using these intervals. However, this representation of the data may not reduce the complexity of the data, especially when the variations are very frequent. Moreover, clustering of similar intervals is a highly complex step, due to the large number of intervals and the sequential nature of the data.

Our proposed method solves these problems by considering only intervals of the data that are of higher importance to the user, instead of all partitions. Additionally, since Hoppner's method used Allen's interval relations [29], it did not take the pattern locations into account and could not, therefore, extract important information, such as delay and direction of the relationships, which are essential for decision making. We solve this problem using concepts such as pattern location and slope.

3. Definitions and Notations

In this section, all air and rammed earth monitoring data can be represented as sequential data.

Definition 1 (Sequence)**.** *A sequence $S = \{s_1, \cdots, s_l\}$ is an ordered set of values, where s_i $(1 \leq i \leq l)$ is a value of a sequence variable, and l is length of the sequence.*

In this regard, each value shows a data point (e.g., in time or space). For each data point, i indicates the index of the data point in the ordered set S. Time-series data are a special case of sequential data where the order is temporal. For example, $S = \{4.6, 4.3, 4.1, 4.4, 7.3, 8.5, 6.6, 5.3\}$ is a daily rammed earth temperature sequence measured every three hours, with l as 8. s_1 is 4.6, which means the rammed earth temperature is 4.6 °C at 0:00, and so on.

Definition 2 (Subsequence)**.** *Sequence $S' = \{s_I, \cdots s_{I+k-1}\}$ is called a subsequence of S if $S' \subseteq S$, and $1 \leq I < \cdots < I+k-1 \leq l$. Here k is the length of the subsequence, and the subsequence S' is an ordered subset of S starting from start index I.*

For example, in sequence S, subsequence $S' = \{s_3, s_4, s_5, s_6\} = \{4.1, 4.4, 7.3, 8.5\}$. I is 3, and k is 4.

Definition 3 (Variance of sequence)**.** *Given a sequence S, the variance of S in term of the first element is defined as:*

$$\delta(S) = \frac{1}{n} \sum_{j=1}^{j=l-1} (s_{j+1} - s_1)^2 \tag{1}$$

Sequence variance measures how far sequence values are spread out. We are interested in the amount of changes from the first element of the sequence (s_1); for sequence $S = \{4.6, 4.3, 4.1, 4.4, 7.3, 8.5, 6.6, 5.3\}$, then $\delta(S) = ((4.3-4.6)^2 + (4.1-4.6)^2 + (4.4-4.6)^2 + (7.3-4.6)^2 + (8.5-4.6)^2 + (6.6-4.6)^2 + (5.3-4.6)^2)/8$ ≈ 3.4. For $S' = \{4.1, 4.4, 7.3, 8.5\}$, $\delta(S') = ((4.4-4.1)^2 + (7.3-4.1)^2 + (8.5-4.1)^2)/4 \approx 7.4$.

Definition 4 (Slope of sequence)**.** *Slope m of a sequence is defined as:*

$$m(S) = \frac{s_l - s_1}{|s_l - s_1|} = \begin{cases} 1, & s_l > s_1 \\ 0, & s_l = s_1 \\ -1, & s_l < s_1 \end{cases}, \tag{2}$$

The slope of a sequence describes the rising or falling direction of the sequence. If the slope is positive, the sequence is rising. If the slope is negative, it is falling. For sequence $= \{4.6, 4.3, 4.1, 4.4, 7.3, 8.5, 6.6, 5.3\}$, *since* $5.3 > 4.6$, *then* $m(S) = 1$. *For subsequence* $S' = \{4.1, 4.4, 7.3, 8.5\}$, *since* $8.5 > 4.1$, *then* $m(S') = 1$.

Definition 5 (Multivariate sequence). *Consider n number of sequence $S_1, S_2, \cdots S_n$. A multivariate sequence Y is a set of sequences denoted as* $Y = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{pmatrix}$. *For index i,* $Y_i = \begin{pmatrix} s_{1i} \\ \vdots \\ s_{ni} \end{pmatrix}$ *where s_{ji} indicates the ith data point of S_j.*

For example, S_1 is a sequence of air temperature and S_2 is a sequence of rammed earth temperature, then $Y = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = \left(\begin{array}{cccccccc} 8.5, & 7.3, & 7.1, & 9, & 15.1, & 14.8, & 11.6, & 8.4 \\ 4.6, & 4.3, & 4.1, & 4.4, & 7.3, & 8.5, & 6.6, & 5.3 \end{array} \right)$ as a multivariate sequence formed from the monitoring data. Multivariate sequential analysis is used to model and explain the interactions and co-movements among a group of sequence variables.

Definition 6 (Interesting pattern). *Given sequence S of length l, subsequence S' of S is interesting pattern p if its variance is greater than a threshold δ_{min}, i.e., $\delta(S') \geq \delta_{min}$.*

In this paper, we are interested in finding the rising and falling patterns in the observed air and rammed earth temperature data. We define the concept of interesting pattern based on the variance.

A group of interesting patterns discovered from a set of sequences is called a candidate pattern set. For example, given $\delta_{min} = 5$, p_{air_3} is an interesting pattern discovered from the air temperature sequence of an earthen ruin, because $S'_{air\_3} = \{10.2,\ 8.8,\ 5.6,\ 3.4,\ 1.2\}$ and $\delta\left(S'_{air\_3}\right) \approx 30.1 > 5$. Given $\delta_{min} = 1$, p_{earth_5} is an interesting pattern discovered from the rammed earth temperature sequence, because $S'_{earth\_5} = \{3.1,\ 2.4,\ 2.1,\ 1.5,\ 0.1\}$ and $\delta\left(S'_{earth\_5}\right) \approx 4.1 > 1$.

Given a set of variables, variables (e.g., air temperature) that can be used for predicting other variables are called conditional variables. The other variables that can be predicted using conditional variables are called decision variables (e.g., rammed earth temperature). In this paper, we study how to predict rammed earth temperature with air temperature, so an important concept is the correlation rule.

Definition 7 (Correlation rule). *Let P be a set of candidate pattern sets related to condition al variables in a dataset of n multivariate sequential data, P' be another set of candidate pattern sets related to decision variables, and P and P' are complement regarding the whole set of candidate pattern sets of the dataset. The correlation rule is defined as the form $r = p \Rightarrow p'$ where $p \in P$ and $p' \in P'$.*

For example, since we focus on prediction of rammed earth temperature with the air temperature of earthen ruins, the correlation rule will be like this $r = p_{air\_3} \Rightarrow p_{earth\_5}$. In this application, earthen ruins conservation experts are interested in not only the combination of the conditional variable patterns influence the decision variables, they are also interested in how the decision variables such as the rammed earth temperature corresponding to every single conditional variable. Therefore, we introduce the following concepts of direction, delay and variation of the rules to describe the possible relationships between single conditional variable and the decision variable. .

Definition 8 (Direction of the rule). *Given rule $r = p \Rightarrow p'$, the direction of rule r is defined as:*

$$D(r) = m(p) \times m(p') = \begin{cases} 1, & m(p) = m(p') = 1 \text{ or } m(p) = m(p') = -1 \\ 0, & m(p) = 0 \text{ or } m(p') = 0 \\ -1, & \text{otherwise} \end{cases}, \tag{3}$$

Here, a positive result means the corresponding variables in the patterns' time intervals move together (positive correlation), while a negative value means opposite movements of the variables in the patterns' time intervals (negative correlation). Otherwise, the direction of the rule is defined as zero.

Definition 9 (Delay of the rule). *Given rule $r = p \Rightarrow p'$, let p.I means the start index value of pattern p, the delay of rule r is defined as:*

$$\Delta(r) = p.I - p'.I \tag{4}$$

The delay explains how long it takes to see the effects of changes of one variable in the value of another variable. For $r = p_{air\_3} \Rightarrow p_{earth\_5}$, start index I of p_{air_3} is 296, start index I of p_{earth_5} is 295, then $\Delta(r) = 296 - 295 = 1$. *If the monitoring frequency is every 3 hours, then $\Delta(r) = 1$ means the delay is 3 h.*

Definition 10 (Variation of the rule). *Given rule $r = p \Rightarrow p'$, let* $\max(p)$ *and* $\min(p)$ *be the minimum and maximum values of the elements in pattern p, the variation of rule r is defined as a pair $(V(p),V(p'))$, where:*

$$V(p) = (\max(p) - \min(p)) \times m(p), \tag{5}$$

$$V(p') = \left(\max(p') - \min(p')\right) \times m(p'), \tag{6}$$

The variation of the rule indicates the degree of changes in terms of one variable in the value of another variable. For example, given rule $r = p_{air\_3} \Rightarrow p_{earth\_5}$, because $S'_{air\_3} = \{10.2,\ 8.8,\ 5.6,\ 3.4,\ 1.2\}$ and $S'_{earth\_5} = \{3.1,\ 2.4,\ 2.1,\ 1.5,\ 0.1\}$, $max(p_{air\_3}) = 10.2$, $min(p_{air\_3}) = 1.2$, $max(p_{earth\_5}) = 3.1$ and $\min(p_{earth\_5}) = 0.1$. *$V(p_{air\_3}) = -9$ and $V(p_{earth\_5}) = -3$; thus, the variation of the rule is (−9, −3).*

4. The PPER Algorithm

In this section, we present the proposed PPER algorithm, which can be viewed as a three-stage process: (1) find interesting patterns; (2) generate predictive rules; and, (3) predict with these predictive rules.

The first stage of PPER is finding interesting patterns that summarize data behavior. In this study, we are specifically interested in variations of the data, so we are looking for rising and falling patterns. Since sequential data may contain repetitive patterns, then the algorithm is to identify similar patterns and group them together. For each group of similar patterns, a representative pattern will be defined. All representative patterns together form a filtered pattern set. The second stage of PPER is using conventional data mining algorithms to retrieve the correlation rules with the filtered pattern set. Then comes filtering and merging, where the correlation rules are then filtered and merged into predictive rules based on earthen ruins prediction requirements. The predictive rules are used to predict the rammed earth temperature of the earthen ruins in the last stage of PPER. The remainder of this section describes the major steps of the algorithm in detail.

4.1. Finding Intersting Patterns

The first stage of PPER algorithm is to find interesting patterns that summarize the data behavior. It includes two steps: (1) identification of the candidate pattern set; and, (2) grouping of similar patterns.

4.1.1. Identifying the Candidate Pattern Set

For each variable in the dataset (e.g., air temperature and rammed earth temperature variables in the earthen ruin monitoring dataset), the algorithm explores all the sequences to find the rising and falling patterns. For each sequence, it starts with a sliding window with a prespecified size from the first data point of the sequence. It then calculates the variance of the subsequence based on Definition 3 and checks whether the variance is greater than a threshold (δ_{min}). If the variance is less than δ_{min}, then there is no interesting pattern in the sliding window and the sliding window is discarded.

Interesting patterns may overlap, for example, the end point of the previous interest pattern is the starting point of a later interesting pattern. Therefore, the partially overlapping window is slid across the sequence and continues with a new sliding window. If the sliding window meets the variance condition, the algorithm continues to extend the sliding window to find the pattern with the maximum possible length. The extension starts by adding the next data point of the sequence to the sliding window. The algorithm continues adding points while the variance keeps increasing and the slope of the last added points is the same as the starting slope. After the extension, the data in the sliding window and its origin information (the identifier of the corresponding sequence) are used to form a pattern. The algorithm then skips those data points and continues finding patterns in the rest of the sequence.

The algorithm is described below. In Algorithm 1, the number of variables of the dataset and the length of each interesting pattern is usually very small, so the time complexity impact caused by them can be ignored. Therefore, the time complexity of the algorithm is $\mathrm{O}(n)$, where n is the number of sequences. Lines 1 to 7 check whether the variance is greater than a threshold (δ_{min}). If the variance is more than δ_{min}, lines 8 to 18 show how to find the longest pattern, and lines 10 to 12 show the details of the extension.

Algorithm 1. FindCandPatterns Algorithm.

FindCandPatterns (Dataset, Window Size, δ_{min}, Overlapping Length)

```
Input: Dataset, Window Size, $\delta_{min}$
Output: candidate pattern set CandSet
1:  Initiate variables and parameters CandSet = ϕ, $w$ = Window Size , o = Overlapping Length;
2:  For each Set $i$ related to variable $i$ in the dataset
3:      For each $S$ in Set $i$
4:          start = 0;
5:          While(start < $S$.length)
6:              SW = $S^{w}_{start}$;   // initiating sliding window
7:              If SW.variance > $\delta_{min}$
8:                  new = SW ∪ $\{s_{\mathrm{SW.end}+1}\}$;   // adding the next data point in S to SW
9:                  while $\delta_{new} > \delta_{SW}$ and slope(new) = = slope(SW)
10:                     SW = SW ∪ $\{s_{\mathrm{SW{\cdot}end}+1}\}$;
11:                 EndWhile
12:                 P = {SW,SID };
13:                 CandSet = CandSet ∪ P;
14:                 start = start + P.length;
15:             Else
16:                 start = start + o;
17:             EndIf
18:         EndWhile
19:     EndFor
20: EndFor
22: Return CandSet
```

4.1.2. Grouping Similar Patterns

After discovering all the candidate patterns of different sequence groups in the dataset, we use these patterns to obtain the hidden relationships between the conditional variable (air temperature in this case) and decision variable (rammed earth temperature). However, for real earthen ruin monitoring datasets with a large number of long sequences, the number of candidate patterns may be very large. As a result, using this collection of patterns for prediction purposes (in the next stage) may not only be computationally expensive in terms of time and space requirements, but also can result in sparse and meaningless predictions. Moreover, the behavior of each sequence may be very similar in different intervals. Such similar behavior may result in the discovery of many similar patterns for one variable. Therefore, we tend to find and filter such similar patterns in the discovered pattern collection to reduce the size of the pattern set and avoid doing repetitive computations for similar patterns. In other words, for every group of similar patterns, we try to determine a representative pattern and use this representative in the next stages of the algorithm.

The first step in finding similar patterns is to generate a distance matrix between patterns. For each pair of patterns, an element of the matrix shows the distance between the two patterns. Therefore, if P is the set of candidate patterns, the distance matrix is a $|P| * |P|$ matrix. A naïve algorithm can be used to calculate the distance between all pairs of patterns. However, the time complexity of generating the matrix is $\mathrm{O}(|P|^2) * \mathrm{O}(f(dist))$, where $|P|$, is the number of patterns identified and $\mathrm{O}(f(dist))$ is the time complexity of the distance function (e.g., dynamic time warping (DTW) or Euclidian distance). It is obvious that the naïve algorithm is highly time consuming, regardless of the distance function, since applying distance measurements on sequential data requires exploring all the data points of the sequences. To tackle this problem we extend and utilize a new data structure based on an R-tree [14] and some pruning rules.

Data Structure

An R-tree is a tree-based data structure that is very popular for indexing spatial data [14]. Unlike Quad-tree, R-tree does not require non-spatial attribute to divide the space and can get better performance without fine tuning the tiling level [37]. The key idea of the data structure is that it represents a group of close data points determined using a minimum bounding rectangle (MBR) and then indexing the MBRs by applying a hierarchical structure. Since the original R-tree was designed for spatial data, it cannot be directly applied to the sequential data. In this study, we modified the R-tree concepts, so that we can use them for indexing subsequences.

Based on the first step of the algorithm, we have the start index I and length n of each pattern p in the candidate pattern set. We also have all the elements of the pattern subsequence. To be able to use the R-tree, the index of the sensor reading is mapped to the x coordinate and the value of the reading to the y coordinate. Therefore, for each pattern p in the candidate pattern set P, we can build an MBR where the corners are $(I, \min(p))$ and $(E, \max(p))$. E is the end index of the pattern p, and $E = I + n - 1$. For example, the start index I of p_1 is 5, the length n of p_1 is 8; therefore, the end index $E = 12$. Figure 4 shows the MBRs of patterns p_1 and p_2 .

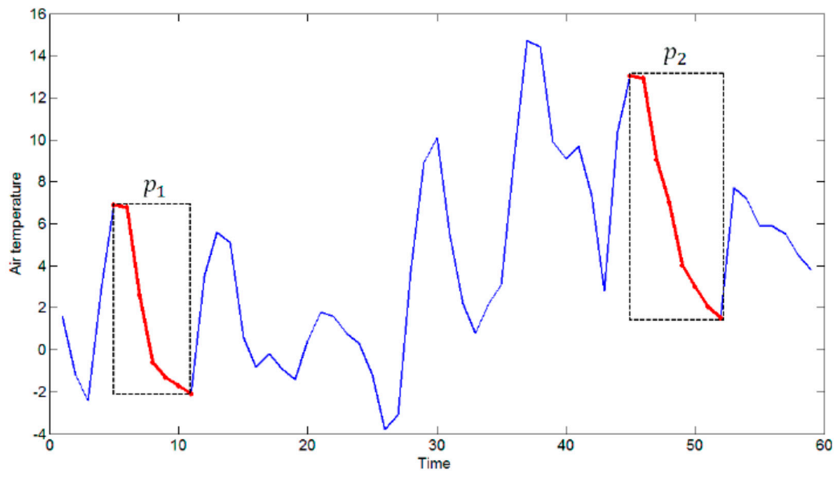


Figure 4. MBR of patterns.

As a result, candidate pattern set P can be indexed using a modified R-tree, as shown in Figure 5, where each leaf is an MBR of a pattern and the intermediate entries of the R-tree index patterns with nearby MBRs.

This new data structure is used for reducing the time complexity by pruning the number of processed patterns.

Definition 11 (Minimum distance between patterns)**.** Given two MBRs of two patterns A and B, the minimum distance between the two patterns is defined as:

$$\mathrm{d}(A,B) = \sqrt{\mathrm{d}_{minX}(A,B)^2 + \mathrm{d}_{minY}(A,B)^2} \tag{7}$$

where $d_{minX}(A,B)$ is defined as the minimum distance between projections of A and B in the x dimension and $d_{minY}(A,B)$ is the minimum distance between projections of A and B in the y dimension.

Figure 6 illustrates an example of the minimum distance for x and y dimensions of two MBRs A and B.

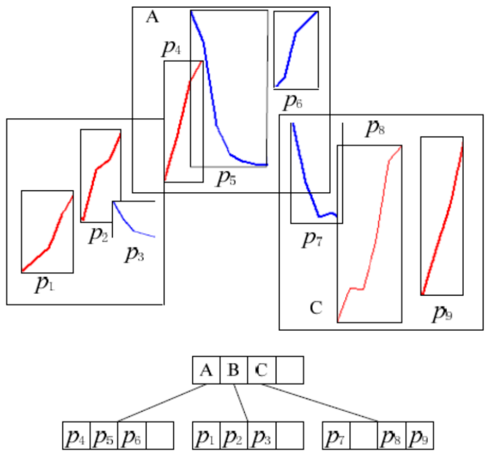


Figure 5. R-tree structure for indexing candidate patterns.

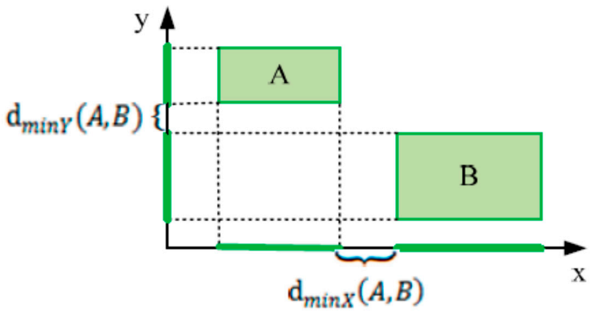


Figure 6. Minimum distance between projections of A and B in x and y dimensions.

Pruning-Based Calculation of Distance Matrix

In order to quickly and efficiently calculate the distance between pattern pairs, we apply the following two pruning rules in the algorithm of pruning-based calculation of distance matrix:

(1) *Pruning rule 1.* If two similar patterns p_1 and p_2 do not appear in the area $(I - W_s, E + W_s)$ of the sequences, then the distance of the two patterns is infinity. W_s is a user specified parameter. Pruning rule 1 is based on the fact that two similar patterns, far from each other, are considered to be two distinct patterns. As a result, there is no need to calculate their distance, and we can simply fill the corresponding matrix element by infinity.

(2) *Pruning rule 2.* If pattern p_1 is falling (negative slope) and pattern p_2 is rising (positive slope), then the distance of the two patterns is infinity. Pruning rule 2 uses for pruning the slope of the patterns. If one pattern is falling (negative slope) and the other one is rising (positive slope) they cannot be considered as similar and therefore we do not need to calculate their similarity. That is, we just simply fill the corresponding matrix element by infinity.

The process of pruning-based calculation of the distance matrix starts from the root node of the tree. For the set of roots' children, it extracts all combinations of two are extracted. For each pair (e.g., ch_1, ch_2), whether the pair elements are patterns or intermediate nodes is first checked. For intermediate nodes, with the first element of the pair (ch_1), and the algorithm of pruning-based calculation of distance matrix is used on the combinations of ch_1's children. The same process is also done for the second element of the pair. We then decide whether we should compare ch_1's children with ch_2's children. To do so, the algorithm calculates the minimum distance between the MBRs of ch_1 and ch_2. If the distance is larger than the distance threshold, it means that the two groups of patterns indexed by the MBRs of ch_1 and ch_2 are either happening far from each other, or the element's values are not close enough. Therefore, we can skip calculating the exact distance between these patterns and fill the corresponding elements of the distance matrix with infinity according to pruning rule 1. In the other case, where the minimum distance between ch_1 and ch_2 MBRs is less than the threshold, the algorithm continues with the combinations of ch_1 and ch_2's children.

The process of pruning-based calculation of the distance matrix proceeds until it reaches the leaves of the tree, which means the elements of the input pairs are patterns. It applies two pruning rules to fill the corresponding element of the distance matrix with infinity. If neither of the pruning rules works, the algorithm calculates the distance between two patterns and fills the corresponding element of the distance matrix. Algorithm 2 details the process for calculating the distance matrix. Lines 2 to 4 show how to get the children list of the root. For each pair (e.g., ch_1, ch_2), line 6 checks whether the pair elements are patterns or intermediate nodes. Lines 7 to 13 show how to fill the corresponding element of the distance matrix according to the two rules.

Algorithm 2. CalDistMatrix Algorithm.

CalDistMatrix (Pattern set P)

```
Input: Pattern set P
Output: distance matrix DistanceMatrix
1:  Initiate DistanceMatrix with infinty;
2:  Rtree patternTree = P.index();
3:  root = patternTree.getRoot;
4:  childrenList = root.getChildren();
5:  For each ($ch_1$,$ch_2$) in childrenList
6:      If $ch_1$,$ch_2$ are not patterns
7:          getChildren($ch_1$);
8:          do
9:          If the children $i$ and $j$ of $ch_1$ are in same area and with same slope
10:             DistanceMatrix[$ch_i$.id][ $ch_j$.id] = dist($ch_i$,$ch_j$);
11:         EndIf
12:         While patterns are found;
13:         getChildren($ch_2$);
14:         do
15:         If the children $k$ and $l$ of $ch_2$ are in same area and with same slope
16:             DistanceMatrix[$ch_k$.id][ $ch_l$.id] = dist($ch_k$,$ch_l$);
17:         EndIf
18:         While patterns are found;
19:         If MBRdist($ch_1$,$ch_2$) < $d_{threshold}$ and $ch_1$ and $_2$ are in same area and with same slope
20:             DistanceMatrix[$ch_1$.id][$ch_2$.id] = dist($ch_1$,$ch_2$);
21:         EndIf
22:     EndIf
23: EndFor
24: Return DistanceMatrix
```

Finding Similar Patterns

After calculating the distance matrix, the algorithm uses the matrix to decide which patterns are similar. For each pair of patterns (p_i, p_j), if their distance is less than distance threshold $\widetilde{d}$, then are similar; and, the more representative one needs to be kept. We consider the one with highest number of similar patterns as the representative one (e.g., p_i) because it covers a wider range of similar patterns and represents a larger population. Therefore, p_j should be added to the set of similar patterns of p_i and then removed from the result list. Continuing this process results in a reduced set of patterns, which can be used for the next step of the algorithm.

Algorithm 3 shows the algorithm to find similar patterns. Line 4 shows which patterns are similar if their distance is less than $\widetilde{d}$. Lines 5 to 12 show how to get the longest pattern to represent other similar patterns.

Algorithm 3. FindSimilarPatterns Algorithm.

```
FindSimilarPatterns( Pattern set P, $\widetilde{d}$)

Input: Pattern set P, $\widetilde{d}$
Output: filteredPatternList
1:  filteredPatternList = P;
2:  For $i$ = 0 to P.size
3:      For $j$ = 0 to P.size
4:          If DistanceMatrix(i,j) < $\widetilde{d}$
5:              If P($i$).similarlist.size() > P($j$).similarlist.size()
6:                  P($i$).similarlist.add(P($j$));
7:                  filteredPatternList.remove(P($j$));
8:              Else
9:                  P($j$).similarlist.add(P($i$));
10:                 filteredPatternList.remove(P($i$));
11:             EndIf
12:         EndIf
13:     EndFor
14: EndFor
15: Return filteredPatternList
```

4.2. Generating Predictive Rules

Since the Apriori algorithm is a classic association rule mining algorithm which is simple, easy to understand and implement, and can obtain the correlation rules between conditional variables and decision variables, so the Apriori algorithm is used it to obtain correlation rules on these interesting patterns. The details of the Apriori algorithm are described in [38] and are not repeated here. Since the rule mining technique does not consider the semantics of the rules, in this paper we apply some filtering as a post-processing step to make sure the rules conform to domain constraints. In this step, based on the application requirements, we define a set of filters for the output rules. In the earthen ruin monitoring dataset, we have a set of patterns discovered in the air temperature data and another set of patterns from rammed earth temperature data. In this case, the rule miner finds all the relationships between patterns regardless of the conditional variable of the patterns. It is then possible to have some rules, such as $r = p_{air\_3} \rightarrow p_{air\_5}$, where the conditional and decision patterns are air temperature data. Since the target of the current study is finding relationships between patterns from air temperature and rammed earth temperature, rules, such as $r = p_{air\_3} \rightarrow p_{air\_5}$, are of no use. Therefore, we can define a filter to remove all the rules that have patterns from the same variable.

We then calculate the direction of each rule. We filter the rules whose directions are -1 or 0, because these rules are often due to the existence of some data anomalies or noise. We also calculate the delay and variation of each filtered rule and merge the above rules into more succinct predictive rules. The predictive rule may be in the following form:

If $\min(p_{air\_i}) > 0\ ^{\circ}\mathrm{C}$ and $+5\ ^{\circ}\mathrm{C} \leq V(p_{air\_i}) \leq +10\ ^{\circ}\mathrm{C}$
then $+2\ ^{\circ}\mathrm{C} \leq V(p_{earth\_j}) \leq +3\ ^{\circ}\mathrm{C}$ with delay = 3 h

4.3. Predicting with Predictive Rules

After obtaining the prediction rule, we can use the time series of conditional variables to predict the time series of decision variables. The patterns are extracted in the time series of conditional variables to match the prediction rules based on the predictive rules. The matching results are used as the predictive results of the time series of decision variables. Hit rate H is defined to show the performance of the proposed algorithm:

$$H = \frac{N}{M} \tag{8}$$

where numerator N is the number of the air temperature patterns that accurately predict the rammed earth temperature pattern, and denominator M. is the number of all the air temperature patterns in the testing dataset.

5. Experiments

To evaluate the proposed PPER method, we applied it to the Ming Dynasty Great Wall dataset monitored from the Ming Dynasty Great Wall in Shaanxi Province of China by an IoT network. We have deployed about 300 IoT nodes in five monitoring areas of the Ming Dynasty Great Wall in Shaanxi. The IoT based sensing infrastructure for the Ming Dynasty Great Wall is shown in Figure 7.

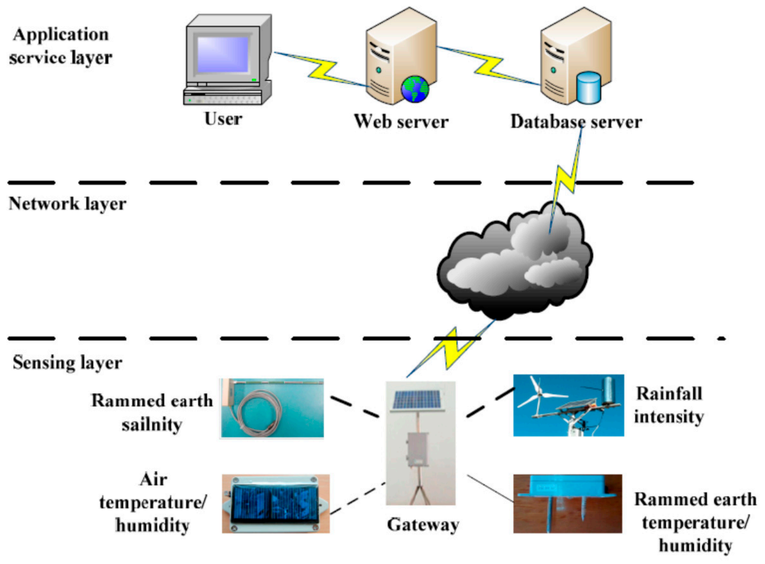


Figure 7. The IoT based sensing infrastructure for Ming Dynasty Great Wall.

The sensing infrastructure includes three layers, sensing layer, network layer and the application service layer. In the sensing layer, there are different IoT nodes which can obtain environmental monitoring data, such as air temperature and humidity, and rainfall intensity, and earthen ruin monitoring data, such as rammed earth temperature, humidity and salinity. With Zigbee transmission technology, these nodes can transmit data to the gateway. In the network layer, with GPRS technology, data are finally reached the database server. A parser is developed to parse the received packets, and save the data into a database using MySQL Server. In the application service layer, user can visit the web server to query the data and view the analysis results. Some photos of the IoT node layouts in the five monitoring areas are shown in Figure 8.

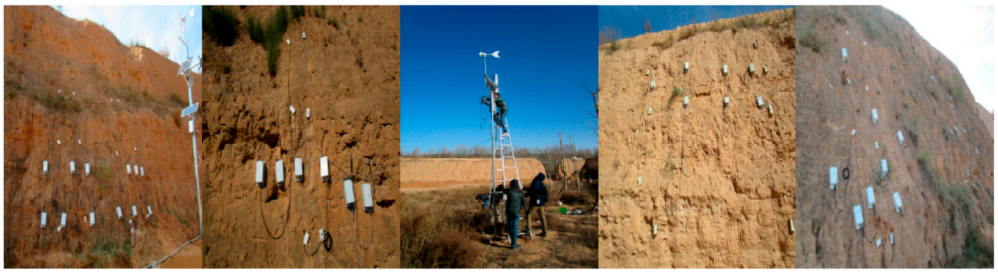


Figure 8. Photos of IoT nodes layouts in the 5 monitoring areas.

In this work, we focused on obtaining rammed earth temperatures; therefore, we used the air temperature, rainfall intensity, rammed earth slope and aspect and rammed earth temperature monitoring data of the Ming Dynasty Great Wall dataset. The dataset consists of the air temperature, the rainfall intensity and the rammed earth temperature underground 5 cm in 5 monitoring areas collected from 29 October 2015 to 29 January 2016. It also consists of the rammed earth slope and aspect of each monitoring area. The air temperature monitoring frequency and the rainfall intensity were once every five minutes, and the rammed earth temperature monitoring frequency was once every three hours. The air temperature, the rainfall intensity and rammed earth temperature underground 5 cm of each monitoring area are considered as a multivariate sequence.

The algorithm was implemented in Java, and all experiments were run on a Windows 7 platform with a CPU speed of 3.4 GHz $\times$ 2, and 12 GB RAM. For the last part of the algorithm, which looks for potential relationships in the data, we used Weka library version 3.7.9.

5.1. Parameter Setting

The window size was set to 4 for the Ming Dynasty Great Wall dataset. The δ_{min} parameter was used to determine whether the algorithm discarded the sliding window or not. For air temperature data, δ_{min} was set to be 5, according to the minimum air temperature difference between day and night. For the rammed earth temperature data, δ_{min} was set to be 1, according to the minimum difference of the rammed earth temperature between day and night. For the rainfall intensity, δ_{min} was set to be 1, according to the minimum rainfall intensity difference between day and night. The overlapping length parameter was also set to 1, because the two adjacent interest patterns had a point overlap. The $\widetilde{d}$ parameter was used to decide which patterns were similar. The smaller the $\widetilde{d}$ parameter, the more the similar patterns, and vice versa. Therefore, $\widetilde{d}$ was set at 18 for the air temperature sequence, 8 for the rammed earth temperature, and 0.1 for the rainfall intensity. Using the Ming Dynasty Great Wall dataset, we designed two sets of experiments to evaluate PPER.

5.2. Experiment on the Performance of Pruning Rules

To prove the effect of pruning rules in reducing the time complexity of the algorithm, we utilized air temperature in the Ming Dynasty Great Wall dataset. In this experiment, we evaluate the performance of FindCandPatterns method using Ming Dynasty Great Wall dataset. The performance of FindCandPatterns Algorithm is showed in Figure 9. As shown in this figure, the time complexity increases linearly with the number of sequences. The experimental results are consistent with the results of our time complexity analysis in Section 4.1.1. We then conducted some experiments to investigate the performance of the distance matrix calculation with and without pruning rules. As Figure 10 shows, the performance of the pruning-based algorithm is much better than the naïve one for both Euclidian and DTW distance measures.

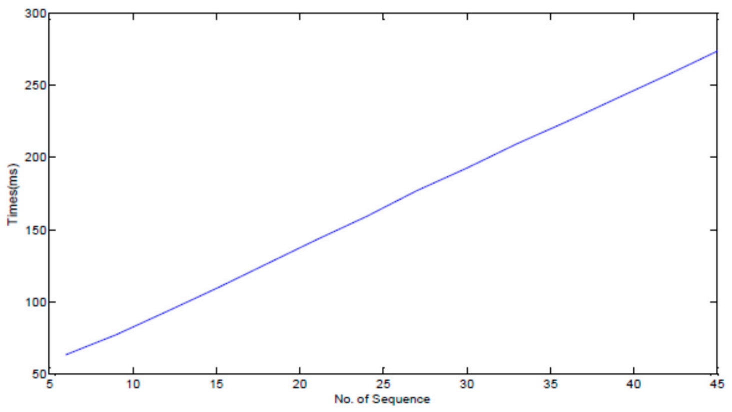
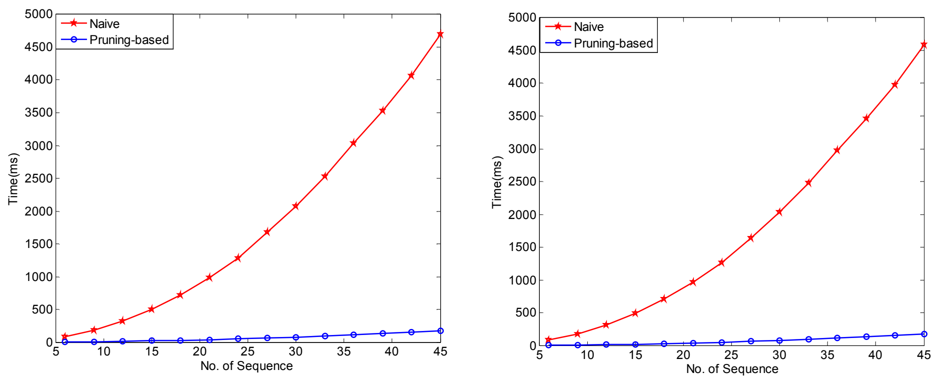


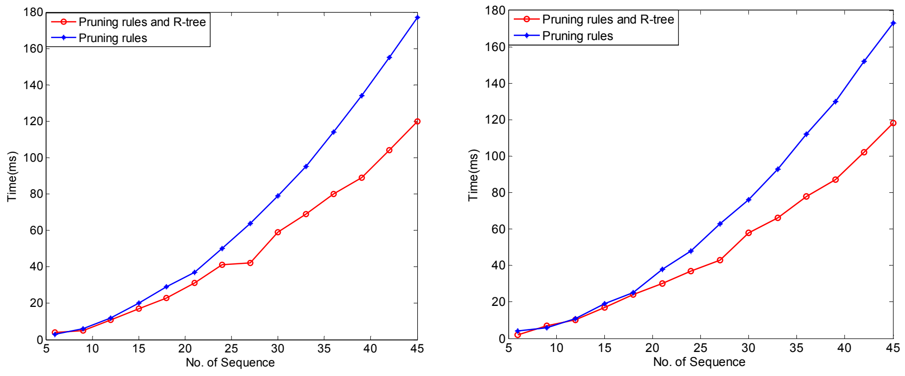
Figure 9. Performance of FindCandPatterns Algorithm**.**



(**a**) Euclidian distance measure (**b**) DTW distance measure

Figure 10. Performance of naïve and pruning-based distance matrix calculation algorithms vs. number of sequences.

Figure 11 presents the results of another experiment—one which studied the effect of two levels of pruning on the performance of the distance matrix calculation algorithm. In the figure, the blue-colored line with ○ shows the algorithm that only applied index and slope pruning rules without using the R-tree-like data structure; and, the red line with * shows the results related to the version of the algorithm that utilizes the R-tree-like data structure for further pruning. It can be seen that for almost all situations, use of the R-tree improved the performance of the calculations. However, for distance matrix calculation, using the R-tree like structure does not lead to a huge performance improvement.



(**a**) Euclidian distance measure (**b**) DTW distance measure

Figure 11. Effect of pruning levels on the performance of distance matrix calculation algorithms.

5.3. Experiments on the Performance of Prediction

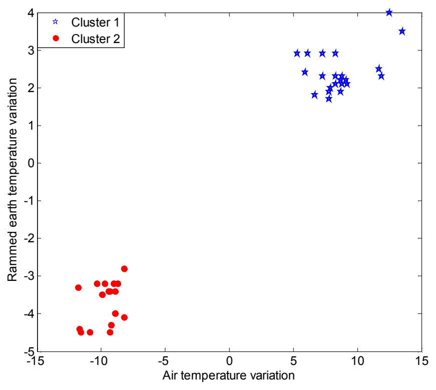
In this work, the Apriori algorithm was used to discover correlation rules from the conditional variables patterns and the decision variable patterns in the Ming Dynasty Great Wall dataset. We selected the monitoring data of the partial nodes with the same orientation of south; half of these data were used to mine the correlation rules, and the other half were used to test the pattern prediction function of the algorithm.

In this experiment, we applied PPER in a dataset with three conditional variables, including air temperature rammed earth slope aspect and rainfall intensity, and obtained 42 the correlation rules. Among all discovered rules, the following example correlation rule describes the relationship among three conditional variables and one decision variable, i.e., the air temperature, rammed earth slope and aspect, and the rainfall intensity, and the rammed earth temperature:

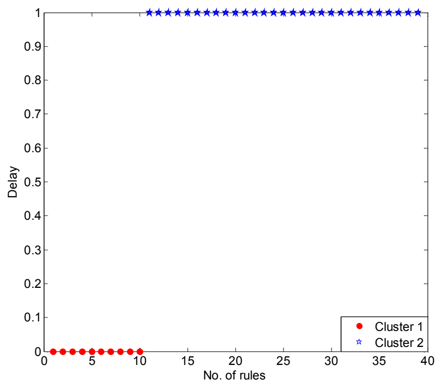
$$r_5 = p_{air\_8}\ ,\ p_{rain\_9}, p_{slope\_1} \Rightarrow p_{earth\_9} \tag{9}$$

where $p_{air\_8} = (7.9,\ 8.5,\ 10.3,\ 11.5,\ 13.9)$, $p_{earth\_9} = (2.1,\ 2.3,\ 2.7,\ 3.5,\ 4.3)$, $p_{slope\_1} = (\ 170)$ and $p_{rain\_9} = (0.01, 0.01,\ 0,\ 0,\ 0)$. It shows when the air temperature increases from 7.9 to 13.9 °C, and the rainfall intensity decreases from 0.01 to 0, the rammed earth slope is toward the south with its aspect of 170 degrees, the temperature of rammed earth increases from 2.1 to 4.3 °C. Because precipitation in our study area is low, so the rainfall intensity data are sparse and does not have much variations, so the number of interesting patterns extracted based on the rainfall intensity are small, which does not provide too much in-depth knowledge to the earthen ruins conservation experts. Therefore, we continue analyzing the rules with air temperature as the conditional variable to predict the rammed earth temperature in order to solve the rammed earth temperature prediction with good performance. After keeping the correlation rules that describe the relationship from the air temperature to the rammed earth temperature and filtering the rules that had a negative direction, we determined 39 correlation rules with a confidence of 0.9.

K-means is a commonly used clustering method which is efficient and fast with the time complexity $O(n)$, where n is the number of data objects. In this application, we use k-means method to analysis the variation and delay of these correlation rules. Because we focus on the two kinds of patterns, falling and rising patterns, so the number of cluster k is set as 2. We obtained the results shown in Figure 12. In Figure 12a, all the rising rules were placed into cluster 1 with air temperature variation from 5.3 °C to 13.5 °C and rammed earth temperature variation from 1.7 °C to 4 °C; and, all falling rules were placed into cluster 2 with air temperature variation from −11.8 °C to −8.2 °C and the rammed earth temperature variation from −4.5 °C to −3.2 °C. Figure 12b shows the two clusters of the delay of the 39 rules. About 74% of the delay was an observation point of time of about 3 h.



(**a**) Rule mining results of variation



(**b**) Rule mining results of delay

Figure 12. Rule mining results for Ming Dynasty Great Wall dataset.

We continued to merge the above rules into more succinct predictive rules, as shown in Algorithm 4. There were six predictive rules that could be used to predict the rammed earth temperature pattern by rule matching.

Algorithm 4. Predictive Rules.

Predictive Rules

```
1: If min($p_{air\_i}$) > 0 °C and +5.3 °C ≤ V($p_{air\_i}$) ≤ +9.2 °C
then +2.1 °C ≤ V($p_{earth\_j}$) ≤ +2.9 °C with delay = 3 h
2: If min($p_{air\_i}$) > 0 °C and −11.7 °C ≤ V($p_{air\_i}$) ≤ −8.9 °C
then − 4.5 °C ≤ V($p_{earth\_j}$) ≤ −4 °C with delay = 3 h
3: If min($p_{air\_i}$) ≤ 0 °C ≤ max($p_{air\_i}$) and +8.8 °C ≤ V($p_{air\_i}$) ≤ +11.9 °C
then +2.1 °C ≤ V($p_{earth\_j}$) ≤ +2.3 °C with delay = 3 h
4: If min($p_{air\_i}$) ≤ 0 °C ≤ max($p_{air\_i}$) and −9.9 °C ≤ V($p_{air\_i}$) ≤ −8.7 °C
then −3.5 °C ≤ V($p_{earth\_j}$) ≤ −3.2 °C with delay = 3 h
5: If max($p_{air\_i}$) ≤ 0 °C and +5.9 °C ≤ V($p_{air\_i}$) ≤ +13.5 °C
then +1.7 °C ≤ V($p_{earth\_j}$) ≤ +3.5 °C with delay = 3 h
6: If max($p_{air\_i}$) ≤ 0 °C and −11.8 °C ≤ V($p_{air\_i}$) ≤ −8.2 °C)
then −3.3 °C ≤ V($p_{earth\_j}$) ≤ −2.8 °C with delay = 3 h
```

The hit rates of the rules are shown in Figure 13. We can observe that predictive rule 4 had the highest average hit rate (89.8%) and that predictive rule 6 had the lowest average hit rate (80.0%). In subsequent analysis, we found that predictive rule 3 merged 10 primitive correlation rules and predictive rule 6 only merged four primitive correlation rules; therefore, predictive rule 3 had the highest average hit rate.

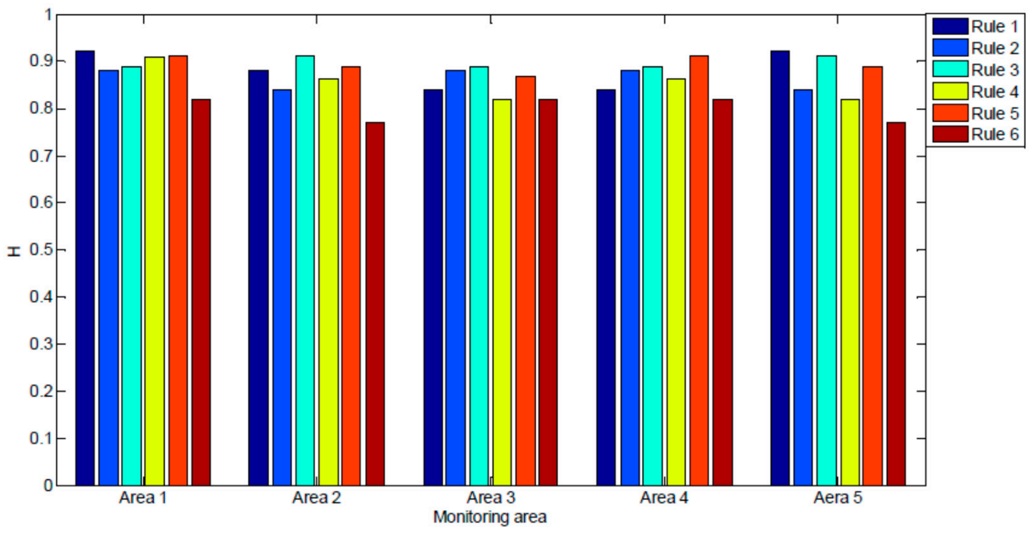


Figure 13. Prediction results for Ming Dynasty Great Wall dataset.

6. Conclusions

In this paper, we investigated the problem of pattern prediction in data obtained from earthen ruins. We proposed the PPER pattern prediction method to find interesting patterns in air and rammed earth temperature sequences. In order to reduce the time complexity of the algorithm, we proposed two pruning rules and a new data structure. Using the correlation rules between the air and rammed earth temperature patterns, predictive rules were obtained. The rammed earth temperature patterns were well predicted with the predictive rules from the Ming Dynasty Great Wall dataset, which has great significance for the protection of the Ming Dynasty Great Wall. We have integrated PPER into the intelligent platform of Ming Dynasty Great Wall, and made a demonstration application in the Ming Dynasty Great Wall in Zhenbeitai, Shaanxi Province, China, and achieved good prediction results, which provided scientific decision-making for the protection of Ming dynasty Great Wall. PPER can be applied on multivariate problems. It can be extended to multiple pattern prediction with similar rising and falling interest patterns, such as the petroleum well log data, to predict the product rate pattern with pressure pattern. In the future, we will collect more monitoring data to improve the prediction performance of PPER. As the same time, we will extend and apply PPER to other multivariate problems. We plan to do more experiments and research to improve the efficiency and effectiveness of PPER. The generalizability of the algorithm will also be a focus of follow-up work, we plan to definite more interesting patterns to solve different pattern prediction.

Acknowledgments: This research was supported by the National Natural Science Foundation of China (No. 61602379), International cooperation and exchange program of Shaanxi Province (No. 2016KW-034), Industrial Science and Technology Project of Shaanxi Province (No. 2015GY013), the Education Department of Shaanxi Province Natural Science Foundation, China (Grant No. 12JK0937, 15JK1742) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

Author Contributions: Yun Xiao and Xin Wang conceived and designed the experiments. Yun Xiao, Faezeh Eshragh and Xuanhong Wang performed the experiments. Yun Xiao and Xin Wang analyzed the data. Xuanhong Wang contributed analysis tools. Yun Xiao wrote the paper. Xiaojiang Chen and Dingyi Fang supervised the overall work and revised the manuscript.

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

References

1. Shu, L. Research of Seismic Stability on Earthen Ruins of Camels City. Master's Thesis, Lanzhou Jiaotong University, Lanzhou, China, 2011.
2. Matero, F. Mud brick metaphysics and the preservation of earthen ruins. *Conserv. Manag. Archaeolog. Sites* **2015**, *17*, 209–223. [CrossRef]
3. Zhao, H.; Li, Z.; Han, W.; Wang, X.; Chen, W. Main diseases and their causes of earthen ruins in arid region of Northwestern China. *Chin. J. Rock Mech. Eng.* **2003**, *22*, 2875–2880.
4. Yao, X.; Zhao, F.; Sun, M. On the temperature variation patterns and forecast model of the enclosed earthen site museum. *Dunhuang Res.* **2014**, *6*, 69–74.
5. Abrardo, A.; Balucanti, L.; Belleschi, M.; Carretti, C.M.; Mecocci, A. Health monitoring of architectural heritage: The case study of San Gimignano. In Proceedings of the IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS), Taranto, Italy, 9 September 2010.
6. Rodriguez-Sanchez, M.; Borromeo, S.; Hernández-Tamames, J. Wireless sensor networks for conservation and monitoring cultural assets. *IEEE Sens. J.* **2011**, *11*, 1382–1389. [CrossRef]
7. D'Amato, F.; Gamba, P.; Goldoni, E. Monitoring heritage buildings and artworks with wireless sensor networks. In Proceedings of the IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS), Perugia, Italy, 28 September 2012.
8. Chianese, A.; Piccialli, F. Designing a smart museum: When cultural heritage joins IoT. In Proceedings of the 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST), Oxford, UK, 10–12 September 2014.
9. Chianese, A.; Piccialli, F. Improving user experience of cultural environment through IoT: The beauty or the truth case study. In *Intelligent Interactive Multimedia Systems and Services*; Springer International Publishing: Cham, Switzerland, 2015; pp. 11–20.
10. Marulli, F. IoT to enhance understanding of cultural heritage: Fedro authoring platform, artworks telling their fables. In *Future Access Enablers of Ubiquitous and Intelligent Infrastructures*; Springer International Publishing: Cham, Switzerland, 2015; pp. 270–276.
11. Alletto, S.; Cucchiara, R.; Del Fiore, G.; Mainetti, L.; Mighali, V.; Patrono, L.; Serra, G. An indoor location-aware system for an iot-based smart museum. *IEEE Internet Things J.* **2016**, *3*, 244–253. [CrossRef]
12. Cuomo, S.; De Michele, P.; Piccialli, F.; Galletti, A.; Jung, J.E. IoT-based collaborative reputation system for associating visitors and artworks in a cultural scenario. *Expert Syst. Appl.* **2017**, *79*, 101–111. [CrossRef]

13. Xiao, Y.; Chen, X.; Wang, L.; Li, W.; Liu, B.; Fang, D. An immune theory based health monitoring and risk evaluation of earthen sites with internet of things. In Proceedings of the IEEE International Conference on Green Computing and Communications (GreenCom) and Internet of Things (iThings) and IEEE Cyber, Physical and Social Computing (CPSCom), Bejing, China, 20–23 August 2013.
14. Guttman, A. R-trees: A dynamic index structure for spatial searching. In Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, Boston, MA, USA, 18–21 June 1984.
15. Zhang, H.; Liu, P.; Wang, J.F.; Wanget, X.D. Generation and detachment of surface crust on ancient earthen architectures. *Rock Soil Mech.* **2009**, *30*, 1883–1891.
16. Sun, B.; Zhou, Z.; Zhang, H.; Zhang, Y.; Zheng, L. Characteristics and prediction model of surface temperature for rammed earthen architecture ruins. *Rock Soil Mech.* **2011**, *3*, 867–871.
17. Cheng, H.; Tan, P.-N.; Gao, J.; Scripps, J. Multistep-ahead time series prediction. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapore, 9–12 April 2006.
18. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*, 5th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2015; pp. 1–17.
19. Holt, C.C. Forecasting seasonals and trends by exponentially weighted moving averages. *Int. J. Forecast.* **2004**, *20*, 5–10. [CrossRef]
20. Brown, R.G. *Smoothing, Forecasting and Prediction of Discrete Time Series*, 1st ed.; Dover Publications: New York, NY, USA, 2004; pp. 97–104.
21. Pegels, C.C. Exponential forecasting: Some new variations. *Manag. Sci.* **1969**, *12*, 311–315.
22. Sodanil, M.; Chatthong, P. Artificial neural network-based time series analysis forecasting for the amount of solid waste in Bangkok. In Proceedings of the Ninth International Conference on Digital Information Management, Phitsanulok, Thailand, 29 September–1 October 2014.
23. Babu, C.N.; Reddy, B.E. A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. *Appl. Soft Comput.* **2014**, *23*, 27–38. [CrossRef]
24. Yoon, H.; Jun, S.C.; Hyun, Y.; Bae, G.O.; Lee, K.K. A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer. *J. Hydrol.* **2011**, *396*, 128–138. [CrossRef]
25. Chen, T.-T.; Lee, S.-J. A weighted LS-SVM based learning system for time series forecasting. *Inf. Sci.* **2015**, *299*, 99–116. [CrossRef]
26. Lin, A.; Shang, P.; Feng, G.; Zhong, B. Application of empirical mode decomposition combined with K-nearest neighbors approach in financial time series forecasting. *Fluct. Noise Lett.* **2012**, *11*, 1250018. [CrossRef]
27. Höppner, F. Knowledge Discovery from Sequential Data. Ph.D. Thesis, Technical University Braunschweig, Braunschweig, Germany, 2003.
28. Lee, A.J.T.; Wu, H.-W.; Lee, T.-Y.; Liu, Y.-H.; Chen, K.-T. Mining closed patterns in multi-sequence time-series databases. *Data Knowl. Eng.* **2009**, *68*, 1071–1090. [CrossRef]
29. Allen, J.F. Maintaining knowledge about temporal intervals. *Commun. ACM* **1983**, *26*, 832–843. [CrossRef]
30. Chakraborty, K.; Mehrotra, K.; Mohan, C.K.; Ranka, S. Forecasting the behavior of multivariate time series using neural networks. *Neural Netw.* **1992**, *5*, 961–970. [CrossRef]
31. Jain, A.; Kumar, A.M. Hybrid neural network models for hydrologic time series forecasting. *Appl. Soft Comput.* **2007**, *7*, 585–592. [CrossRef]
32. Han, M.; Fan, M. Application of neural networks on multivariate time series modeling and prediction. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006.
33. Moerchen, F. Algorithms for time series knowledge mining. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006.
34. Singh, S. Noisy time-series prediction using pattern recognition techniques. *Comput. Intell.* **2000**, *16*, 114–133. [CrossRef]
35. Alvarez, F.M.; Troncoso, A.; Riquelme, J.C. Energy time series forecasting based on pattern sequence similarity. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1230–1243. [CrossRef]
36. Mörchen, F.; Ultsch, A. Efficient mining of understandable patterns from multivariate interval time series. *Data Min. Knowl. Discov.* **2007**, *15*, 181–215. [CrossRef]

37. Kothuri, R.K.V.; Ravada, S.; Abugov, D. Quadtree and R-tree indexes in oracle spatial: A comparison using GIS data. In Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, WI, USA, 2–6 June 2002.
38. Han, J.; Han, I.; Micheline, K. *Data Mining: Concepts and Techniques*, 2nd ed.; Elsevier: San Francisco, CA, USA, 2006; pp. 234–272.