



Temporal Dependency Mining from Multi-sensor Event Sequences for Predictive Maintenance

Weiwei Cao^{1,2(✉)}, Chen Liu^{1,2}, and Yanbo Han^{1,2}

¹ Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, North China University of Technology, Beijing, China

m13611031336@163.com, {liuchen, hanyanbo}@ncut.edu.cn

² School of Computer Science,

North China University of Technology, Beijing, China

Abstract. Predictive maintenance aims at enabling proactive scheduling of maintenance, and thus prevents unexpected equipment failures. Most approaches focus on predicting failures occurring within individual sensors. However, a failure is not always isolated. The complex dependencies between different sensors result in complex temporal dependencies across multi anomaly events. Therefore, mining such temporal dependencies are valuable as it can help forecast future anomalies in advance and identifying the possible root causes for an observable anomaly. In this paper, we transform the temporal dependency mining problem into a frequent co-occurrence pattern mining problem and propose a temporal dependency mining algorithm to capture temporal dependency among multi anomaly events. Finally, we have made a lot of experiments to show the effectiveness of our approach based on a real dataset from a coal power plant.

Keywords: Predictive maintenance · Root causes · Temporal dependency · Frequent co-occurrence pattern

1 Introduction

Predictive maintenance aims to help anticipate equipment failures to allow for advance scheduling of corrective maintenance. It is usually performed based on an assessment of the health status of equipment [1, 2]. Thanks to the rapid development of IoT, massive sensors are deployed on industrial equipment to monitor health status. As a key phase for predictive maintenance, anomaly detection technologies have given us the ability to monitor anomalies from multivariate time-series sensor data or events [3].

In recent years, lot of researches have paid attention to the problem of anomaly detection/prediction from multivariate time-series sensor data. Anomaly detection within individual variables, referred to as “univariate anomaly”, have already been extensively studied [4, 5]. However, it is much more challenging but common in the real applications to mine and analyze temporal dependencies among sensor data or “univariate anomaly” events. It means the possibility of finding new anomaly type or inferring the root cause of an anomaly [6–8].

In Sect. 2, a simple example shows the temporal dependencies found among “univariate anomaly” events. Actually, in a complex industrial system, an anomaly/failure is not always isolated. Owing to the obscure physical interactions, trivial anomalies will propagate among different sensors and devices, and gradually deteriorate into a severe one in some device [9]. Mining such temporal dependencies are valuable as it can help forecast future anomalies/failures in advance and identifying the possible root causes for an observable device anomaly/failure [10].

In the paper, we try to propose an effective and explainable approach to predict the anomaly based on mining the temporal dependencies from multi-sensor event sequences. To reach this goal, we detect “univariate anomaly” events from sensor data and output multi-sensor event sequences. Then, we transform the temporal dependency mining problem into a frequent co-occurrence pattern mining problem. Next, a graph-based anomaly prediction model is built based on choosing and connecting the mined temporal dependencies for event predict in the experiment. Furthermore, a lot of experiments are done to show the effectiveness of our approach based on a real dataset from a coal power plant.

2 Motivation

An anomaly event carries much information about an anomaly like its occurrence time, sources and type. Here, we use a 4-tuple to depict an anomaly event: $e = (timestamp, eventid, sourceid, type)$, where *timestamp* is the occurrence time of e ; *eventid* is the unique identifier of e ; *sourceid* is the unique identifier of the source sensor; and *type* is the type of anomaly event.

A time-ordered list of events from the same sensor construct an event sequence $E_i = \{e_1, e_2, \dots, e_m\}$. All the event sequences construct the event space $\Theta = \{E_1, E_2, \dots, E_n\}$. These events are not isolated with each other. They imply complex temporal dependencies.

Definition 1. Temporal Dependency: Let $A = \{E_1, E_2, \dots, E_k\}$, $1 \leq k < m$ be an event set contains k event sequences, $B = \{E'_1, E'_2, \dots, E'_h\}$ $1 \leq h < m$ be another event set contains another h event sequences, and $A \cap B = \emptyset$. A temporal dependency [10, 11] is typically denoted as $A \xrightarrow{[t_1, t_2]} B$. It means that B will happen within time interval $[t_1, t_2]$ after A occurs.

Figure 1 shows a sample about temporal dependency among several anomaly events. There are several types of anomaly events have occurred on the different sensors. These events construct five event sequences which are shown in Fig. 1. In which, the event sequence $E^1 = \{e_i^1, i = 1, 2, 3, 4, 5, 6, 7\}$ is constructed by H-CF events, and the event sequences $E^3 = \{e_j^3, j = 1, 2, 3, 4, 5\}$ is constructed by the H-IAP event. According to

the maintenance log, when every event e_j^3 in E^3 happen, the event e_i^1 in E^1 will happen within an average time lag $\Delta t = 18$ min. Thus, there is a temporal dependency between the H-CF event and H-IAP event, i.e. $H - CF \xrightarrow{18min} H - IAP$. Thus, if there is a H-CF event, we can predict that in the flowing 18 min, there will happen the H-IAP event.

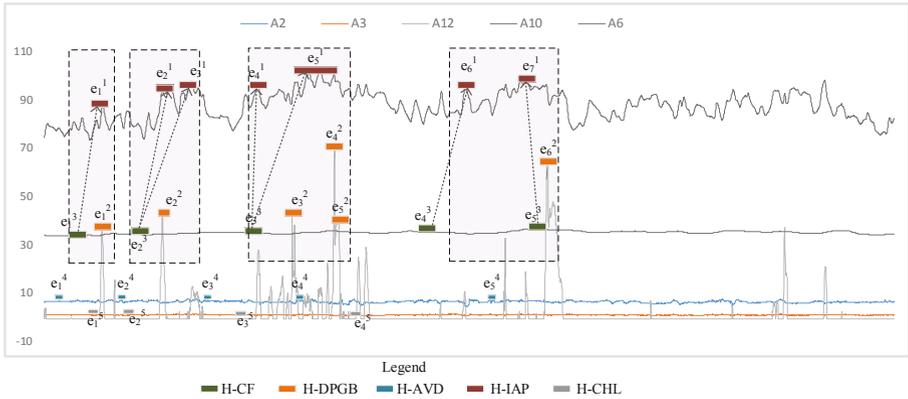


Fig. 1. A real case: Temporal dependencies among event sequences.

Besides the temporal dependency among two event sequences, there is temporal dependency among multi sequences. For example in Fig. 1, the part covered by a shadow contains three event sequences, $E^1, E^2 = \{e_k^3, k = 1, 2, 3, 4, 5, 6\}$ which is constructed by the H-DPGB events and E^3 , when events in E^1 and E^3 occurs, the event in E^2 also occur within an average time lag $\Delta t = 61$ min. Thus, we can see that there is a temporal dependency between the H-CF event, H-IAP event and H-DPGB event, donated as $(H - CF, H - IAP) \xrightarrow{61min} H - DPGB$.

This case illustrates that if we can discovery such temporal dependency for multi events, we have chances of predicting the anomaly event or inferring the root cause of an observable anomaly. Thus, the goal of this paper is mining the temporal dependency from multi event sequences.

3 Temporal Dependency Mining

3.1 Overview

For now, there are lots of excellent techniques have been developed to detect the univariate events. The common ones include range-based approaches, outlier detection approaches [4, 5]. A range-based approach customizes value bounders for individual sensor based on inspectors’ experiences, sensor/device instructions and so on. Outliers are widely known as the values which sufficiently deviate from most ones, the original outlier detection methods were arbitrary, but in recent years statistics techniques are used [12].

The main idea of mining the temporal dependency is to transform temporal dependency into a frequent co-occurrence pattern across multi event sequences. Essentially, a temporal dependency among the means that an event set B frequently occurs within time interval Δt after an event set A occurs. In other words, the temporal dependency is a relationship among the objects in a frequent co-occurrence pattern within a time interval. It inspires us to mine frequent co-occurrence patterns so as to discover event temporal dependencies. This process will be detailed described in Sects. 3.2 and 3.3.

3.2 The Frequent Co-occurrence Pattern Mining

In this section, we explain what a frequent co-occurrence pattern across multi event sequences is, what the differences between the novel pattern mining and traditional frequent co-occurrence pattern mining are, and how to mine the novel patterns. We first list some related concepts in mining the frequent co-occurrence patterns.

Co-occurrence Pattern: For a set of objects $\mathcal{O} = \{o_1, o_2, \dots, o_k\}$ that appear in the same event sequence E_i , an object refers to an event type. $T(\mathcal{O})^{E_i} = \{t_{o_1^{E_i}}, t_{o_2^{E_i}}, \dots, t_{o_k^{E_i}}\}$, $t_{o_j^{E_i}}$ is the occurrence time of o_j ($j = 1, 2, \dots, m$) in E_i , if the \mathcal{O} satisfies that $\max(T(\mathcal{O})^{s_i}) - \min(T(\mathcal{O})^{s_i}) \leq \xi$, then we say that \mathcal{O} is a co-occurrence pattern (CP), ξ is a user-specified threshold.

However, the challenge is how to identify the time lag between two event sets who has the temporal dependency. It actually reflects how long that a set of events will be affected by its related events. Unfortunately, most traditional frequent co-occurrence pattern mining algorithms cannot directly solve such problem. They only focused on the occurrence frequency of a group of unordered objects [13]. Hence, we try to design an algorithm to discover a constrained frequent co-occurrence pattern. Such pattern consists of two object groups, where intra-group objects are unordered and inter-group objects are time-ordered, and all objects span no more than Δt . We call such pattern as frequent co-occurrence pattern across multi event sequences.

Frequent Co-occurrence Pattern Across Multi Event Sequences: For a Co-occurrence pattern $\mathcal{O} = \{\mathcal{O}_{pre} \cup \mathcal{O}_{post}\}$ that occurs in a set of l event sequences $\{E_1, E_2, \dots, E_l\}$. The \mathcal{O}_{pre} and \mathcal{O}_{post} will form the multi-dimensional co-occurrence pattern, denoted as $MCP(\mathcal{O}_{pre}, \mathcal{O}_{post})$, if the \mathcal{O} satisfies the following conditions: (1) every object $o_i \in \mathcal{O}_{pre} \cup \mathcal{O}_{post}$ comes from different event sequences; (2) the object in \mathcal{O}_{post} always occurs after the occurrence of object in \mathcal{O}_{pre} ; (3) $\max\{T(\mathcal{O}_{post})\} - \min\{T(\mathcal{O}_{pre})\} \leq \Delta t$, in which Δt is the time lag, \mathcal{O}_{pre} contains m events and \mathcal{O}_{post} contains n events, i.e. $|\mathcal{O}_{pre}| = m$ and $|\mathcal{O}_{post}| = n$. Thus, the $MCP(\mathcal{O}_{pre}, \mathcal{O}_{post})$ also can be denoted

as $MCP_{m,n}(O_{pre}, O_{post})$. If the $MCP_{m,n}(O_{pre}, O_{post})$ have occurred more than k times in l event sequences $\{E_1, E_2, \dots, E_l\}$, then the $MCP_{m,n}(O_{pre}, O_{post})$ will be regarded as the Multi-dimensional Frequent Co-occurrence Pattern, denoted as $FMCP(O_{pre}, O_{post})$ or $FMCP_{m,n}(O_{pre}, O_{post})$. In which, O_{pre} is the antecedent, O_{post} is the consequent, Δt is the time lag between them.

Based on the above definition, it is obviously that, our FMCP mining task is significantly different from the traditional one. The difference is a MCP is supposed to be divided into two groups, where intra-group objects are unordered and inter-group objects are time-ordered. This time constraint raises the complexity of our task. Assume that the frequency of a FMCP $\mathcal{O} = \{o_1, o_2, \dots, o_m\}$ is l . To find out all valid divisions by traditional ideas, we have to count the frequency for any possible division of \mathcal{O} . The number of possible divisions is $2 * (C_m^2 + \dots + C_m^{\lceil m/2 \rceil})$, where $\lceil m/2 \rceil$ will return the closest integer greater than or equal to $m/2$, not to mention the number of object groups. Owing to this difference, our task is unable to be simply solved by the well-known generation and counting strategy.

3.3 FMCP Mining Algorithm

In this paper, we use $\gamma(A, B)$ to denoted the temporal dependency between event sequences, and we use the $\gamma(A, B).sup$ to donate the occurrence probability of B given the knowledge that A have occurred. It is used to filter the mined temporal dependency.

$$\gamma(A, B).sup = sup(O_{post}|O_{pre}) = \frac{freq(O_{post}|O_{pre})}{freq(O)}$$

In which, $freq(O_{post}|O_{pre})$ is the frequency of the occurrence of O_{post} after with O_{pre} , and $freq(O)$ is the frequency of O .

Assume that the occurrences threshold $freq_{min} = sup_{min}$, where sup_{min} is the threshold of support, and assume that FP is the set constructed by all the FMCP in Θ . All temporal dependency relationships whose satisfied that $sup > sup_{min}$ constitute the set R, then $\forall \gamma(E_i, E_j) \in R$ and there is and only one $TFCP(O_{pre}, O_{post}) \in FP$ satisfy that $\gamma(E_i, E_j)$ is the temporal dependency of O_{pre} and O_{post} , and vice versa.

Because of that $\gamma(E_i, E_j) \in R$, so $\gamma(E_i, E_j).sup \geq sup_{min}$, thus, the number of occurrences of E_j is over sup_{min} after the occurrence of E_i within a time range $\gamma(E_i, E_j).\Delta t$. In a conclusion, E_i and E_j be a FMCP, denoted as $TFCP(E_i, E_j)$, i.e. $TFCP(E_i, E_j) \in FP$. Every FMCP is constructed with antecedent and the consequent, and the item is the set is unique. Therefore, if there is $MCP(O'_{pre}, O'_{post}) \neq FMCP(O_{pre}, O_{post})$ and $FMCP(O'_{pre}, O'_{post}) \in P$, which satisfy that $\gamma(E_i, E_j)$ is the temporal dependency of O'_{pre} and O'_{post} ,

then $O'_{pre} = E_i, O'_{post} = E_j$, and the number of occurrences is $freq(FMCP(O'_{pre}, O'_{post})) = \gamma(E_i, E_j).sup$, and time lag is $FMCP(O'_{pre}, O'_{post}).\Delta t = \gamma(E_i, E_j).\Delta t$. Then $TFCP(O'_{pre}, O'_{post}) = FMCP(O_{pre}, O_{post})$, but this is conflict. Thus, it is proofed that $\forall \gamma(E_i, E_j) \in R$, there is and only one $FMCP(O_{pre}, O_{post}) \in FP$ satisfy that $\gamma(E_i, E_j)$ is the temporal dependency of O_{pre} and O_{post} .

Therefore, if we can get all the FMCP in Θ , we can calculate support of temporal dependencies and filter the candidate temporal dependency sets that satisfy the conditions. Thus, based on the traditional method of frequent co-occurrence pattern mining, in this paper, we proposed an approach called as *ETD-mining* with a three-stage process, that is “Generation-Filter-Extension”, to mine the FMCP.

Given a Frequent Co-occurrence Pattern $FMCP(O_{pre}, O_{post})$, the frequent number is donated as $freq(FMCP(O_{pre}, O_{post}))$, and the $freq(MCP(O_{pre}, O_{post})) = \gamma(O_{pre}, O_{post}).sup$. If the $freq_{min} = sup_{min}$, then for any temporal dependency $\gamma(E_i, E_j)$ that satisfied $\gamma(E_i, E_j).sup > sup_{min}$, there is $freq(MCP(E_i, E_j)) \geq freq_{min}$, it means that $MCP(E_i, E_j)$ is a FMCP. Thus, for any event $e \in E_i \cup E_j$, the occurrence number in some sequence satisfy $freq(e) \geq freq_{min}$. Therefore, the first step of mining FMCP is to find all the events whose occurrence number over $freq_{min}$, these events denoted as F^1 . This step is consistent with the traditional method of frequent co-occurrence pattern mining.

According to the analysis of the above, any support more than the threshold value of event correlation, which incorporates the event set and the target event set can only consist of the event in F^1 . Therefore, we can get the source through the events in combination F^1 events set and target set, and then to filter out support more than the threshold value of event temporal dependency.

The more the event sequences, the higher the cost of filter. If there is n_{seq} event sequences in Θ , and every event sequence contains m_i types of events, then in Θ there are $\dot{m} = \sum_{i=1}^{i=n_{seq}} m_i$ types of events. These events construct the temporal dependency sets can be denoted as $canP = \{\langle E_i, E_j \rangle\}$, then the number of $canP$ is $|canP| = C_{\dot{m}}^2 * C_2^1 + \dots + C_{\dot{m}}^k * (C_k^1 + \dots + C_k^{k-1}) + \dots + C_{\dot{m}}^{\dot{m}} * (C_{\dot{m}}^1 + \dots + C_{\dot{m}}^{\dot{m}-1})$. And for any $\langle E_i, E_j \rangle \in canP$, it is need to verify that if $\gamma(E_i, E_j).sup \geq sup_{min}$. It is obviously that the temporal dependency number need to be filter is very large.

Thus, we designed an extension strategy to avoid this problem based on the following Theorem.

Theorem 1: For any $FMCP(E_i, E_j)$ and the temporal dependency $\gamma(E_i, E_j)$ of event E_i, E_j . Assume that the frequency threshold is equal to the support threshold, i.e. $freq_{min} = sup_{min}$, then for any subset $E'_i \subseteq E_i$ and $E'_j \subseteq E_j$, they also be $FMCP(E'_i, E'_j)$. Besides, when they satisfied that the time lag of temporal dependency $\gamma(E'_i, E'_j) \cdot \Delta t \geq \gamma(E_i, E_j) \cdot \Delta t$, then there is $\gamma(E'_i, E'_j) \cdot sup \geq \gamma(E_i, E_j) \cdot sup$.

Proof: It is obviously that the occurrence count number of E_j is $\gamma(E_i, E_j) \cdot sup$ within a time range $\gamma(E_i, E_j) \cdot \Delta t$ after the occurrence of E_i . For that $E'_i \subseteq E_i$ and $E'_j \subseteq E_j$, thus, the E'_j will occur at last $\gamma(E_i, E_j) \cdot sup$ times in a time range $\gamma(E_i, E_j) \cdot \Delta t$ after the occurrence of E'_i .

Based on the Theorem 1, we can inference that for any $\gamma(E_i, E_j)$ (the support has over the threshold), assume that $|E_i| > 1$, $|E_j| > 1$, the we can get the $FMCP(E_i, E_j)$ of $\gamma(E_i, E_j)$ by extending some $FMCP(\{\varepsilon_\alpha\}, \{\varepsilon_\beta\})$, in which $\varepsilon_\alpha \in E_i$ and $\varepsilon_\beta \in E_j$. We can compose the event in F^1 to get all the patterns such as $(\{\varepsilon_\alpha\}, \{\varepsilon_\beta\})$, then verify that if the pattern satisfied $\gamma(\{\varepsilon_\alpha\}, \{\varepsilon_\beta\}) \cdot sup \geq sup_{min}$, and filter the temporal dependencies whose support over the threshold. Thus, we can get $FMCP(\{\varepsilon_\alpha\}, \{\varepsilon_\beta\})$. Then, choose the remaining event in F^1 to extend the antecedent and consequent of $FMCP(\{\varepsilon_\alpha\}, \{\varepsilon_\beta\})$. At the same time, to verify that if the supports of temporal dependencies for antecedent and consequent are more than the threshold.

Algorithm 1 is the pseudocode of mining the temporal dependency. It first finds the events who had occurred more that sup_{min} times from the event space Θ , and put them into the F^1 (line 1–2). Then, modeling any two events into the pattern $(\varepsilon_\alpha, \varepsilon_\beta)$, and verifying that if the formed pattern is $TFCP_{1,1}$ (line 3–6). Next, it uses the *extend* () function (line 18–26) to expand antecedents of $FMCP_{1,1}$ recursively (line 7–8). The extension process will break up until the extended pattern is not FMCP or there is no object in F^1 can be extended. Based on the result of extension for antecedents, we use *extend* () function to expand consequents of $FMCP_{1,1}$ recursively (line 9–11). Finally, we put all the FMCP into the set P, the event relationship between antecedents and consequents components in P constitute the set R.

Algorithm1: ETD-mining

Input: Θ , event space which contains multi event sequences

$\delta_t \in [\Delta t_{min}, \Delta t_{max}]$, the time lag span

sup_{min} , threshold of support

Output: R , all significant temporal dependencies over Θ

1. for each $seq_i \in \Theta$ // Generation stage
2. put the events into sup_{min} //whose occurrence number over sup_{min} ;
3. for each $\varepsilon_\alpha, \varepsilon_\beta \in F^1$
4. initialize the $E_i = \{\varepsilon_\alpha\}$, $E_j = \{\varepsilon_\beta\}$ and $P = \phi$;// P is FMCP
5. if(is FMCP(E_i, E_j, δ_t)) //Filter Stage
6. $P \leftarrow TFCP(E_i, E_j)$;
7. for each $\varepsilon \in F^1$ && $\varepsilon > E_i$ // ε is greater than E_i
8. extend ($E_i, \varepsilon, \delta_t$); // Extension Stage, extend antecedents
9. for each FMCP(E_p, E_q) $\in P$
10. for each $\varepsilon'' \in F^1$ && $\varepsilon'' > E_j$
11. extend ($E_q, \varepsilon'', \delta_t$); // extend consequents
12. For each FMCP(E_u, E_v) $\in P$
13. $R \leftarrow (E_u, E_v, FMCP(E_u, E_v). \Delta t, FMCP(E_u, E_v). freq)$; // The antecedents and consequents components of FMCP in P constitute the set R
14. return R ;
15. isFMCP(E_i, E_j, δ_t); // judge that if E_i and E_j construct the FMCP(E_i, E_j) with the condition of δ_t ;
16. if ($FMCP(E_i, E_j). \Delta t = \max\{MCP(E_i, E_j). \Delta t\}$ && $FMCP(E_i, E_j). \Delta t \leq \delta_t$)
17. return true;
18. extend ($(E_i, E_j), target, \varepsilon, \delta_t$):
19. IF target == E_i

```

20.  $E_i \leftarrow \varepsilon;$ 
21. IF  $target == E_j$ 
22.  $E_j \leftarrow \varepsilon;$ 
23. IF (isTFCP ( $E_i, E_j, \delta_t$ ))
24.  $P \leftarrow TFCP(E_i, E_j);$ 
25. for each  $\varepsilon' \in F^1 \&\& \varepsilon' > E_i$ 
26. extend ( $(E_i, E_j), target, \varepsilon', \delta_t$ );

```

4 Experiments

4.1 Experiment Dataset and Environment

Environments: The experiments are done on a PC with four Intel Core i5-6300HQ CPUs 2.30 GHz and 16.00 GB RAM. The operating system is Centos 6.4. All the algorithms are implemented in Java with JDK 1.8.5.

Dataset: There are totally 361 sensors deployed on 8 important devices. Each sensor generates one record per second, the dataset size is about 3.61 GB. And the dataset is divided into the training dataset and test dataset. The training set is from 2014-10-01 00:00:00 to 2015-03-31 23:59:59. The testing set is from 2015-04-01 00:00:00 to 2015-04-30 23:59:59.

We use real faults contained in the maintenance records of the plant power from 2014-07-01 00:00:00 to 2015-06-30 23:59:59 to verify our warnings.

4.2 Experiment Setup

Firstly, we conduct the temporal mining algorithm on the training dataset to find the temporal dependencies. We can get the temporal dependency quantity (TDQ) of the training data. We observe the variation trend of the TDQ under the different parameters of algorithm and different time of dataset.

Then, we built an anomaly prediction model with a directed graph over the anomaly events based on choosing and connecting their mined temporal dependencies. The graph is defined as $G = \langle V, E \rangle$, where V is the set of anomaly events, and $E \subseteq V \times V$ is a non-empty set of edges. Each direct edge $v_i \rightarrow v_j$ along with a weight, the weight is the time lag Δt , it means that if the anomaly event v_i occurs, we can predict that the anomaly event v_j will occur after a time lag Δt .

Based on the anomaly prediction graph, we conduct the experiment on test data set. We compare our anomaly prediction model with the other two typical approaches, they are the range-based approach [4], the outlier detection approach [5]. Once the associated anomalies are detected by them, they will make a warning of maintenance to the corresponding fault.

For evaluations, we consider the following performance metrics.

The Temporal Dependency Quantity (TDQ): The Number of temporal dependencies mined in the input data set;

Warning Time: Warning time is the difference between the timestamp an approach makes a warning of maintenance for a fault and the starting time of this fault;

Precision: Precision represents how many abnormalities are accurate according to failure records;

Recall: It presents the probability of being able to classify positive cases, which is defined as following.

4.3 Experiment Result

Firstly, we conduct the experiment to verify how the value of TDQ changes under different data size with different time range when the parameter $sup_{min} = 0.8$ and $\Delta t = 10$. The data size increase from 1-month data to 6-months (the whole training set) dataset and each time for 1 month. This experiment mining the temporal dependences with no less than 0.8 probability (i.e., $sup_{min} = 0.8$). Table 1 shows the result of TDQ.

Table 1. Experiment results of TDQ under different data size ($sup_{min} = 0.8$, $\Delta t = 10$ min).

Time range	Data size	TDQ
2014.07 (1 month)	180 MB	4274
2014.07-2014.08 (2 months)	378 MB	4338
2014.07-2014.09 (3 months)	534 MB	4269
2014.07-2014.10 (4 months)	750 MB	4407
2014.07-2014.11 (5 months)	912 MB	4430
2014.07-2014.12 (6 months)	1128 MB	4954

The Table 1 shows that as the size of the data set increases, the total TDQ also shows an upward trend. However, it is clearly that there is no linear correlation between the TDQ and the size of the data set. Overall, the size of the data set has risen from one month to six months, and the TDQ generated is relatively close, always between 4,000 and 5,000. It is indicated that, as the time goes on increases, the TDQ does not rise rapidly, but increases relatively slowly and slowly, and may even be controlled within a certain range. The reason is that as the time goes on, the more temporal dependencies gradually enriched, but the growth rate trend is stable. This result indicated that our algorithm has a certain robustness.

Then, we compare the precision results and recall results of different methods. Notably, in this paper, we only consider the predict events with finally failures occurring both in training set and testing set. Figure 2 shows the final average results.

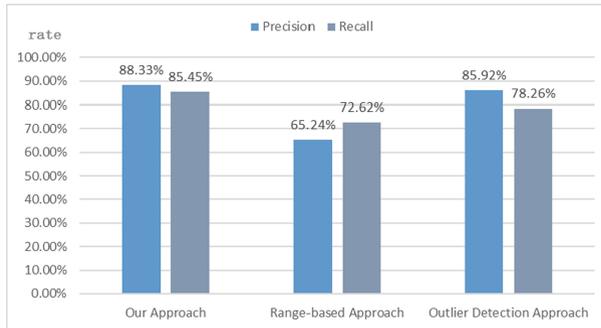


Fig. 2. Experiment Results of Precision and Recall under different approach

Figure 2 indicates that our methods performs well among the three methods, and achieves the highest accuracy and the second highest recall rate.

As shown in the Fig. 2, the precision of our approach is 88.33%, and the recall rate is 85.48%. The analysis of the intermediate results revealed that some of the repetitive anomaly propagation paths excavated in the training cannot be detected in the test set. For the same type of faults, the abnormal propagation path excavated in the test data set has changed compared to the training set. This phenomenon is essentially caused by caused by the character of uncertainty in the stream data.

The precision of the ranged-based approach and the outlier detection approach are 65.24% and 85.92%, and the recall of them are 72.62% and 78.26%. Analysis of the intermediate results revealed that the two approaches are based on the single sensor data. However, usually one fault may be caused by multiple anomalies, it can correspond to multiple isolated anomalous points. The signal sensor data-based detection method cannot detect such a fault. Our approach will find the correlation between multi sensors and then form the anomaly propagation paths, this helps us to find more hidden anomalies.

The above results show that the temporal dependency has an effective effect in constructing fault prediction logic and fault detection.

5 Related Works

Recently, researchers have designed several approaches dealing with the problem of anomaly prediction for the predictive maintenances. Several quantitative models ranging from simple linear discriminant analysis, more complex logistic regression analysis, and neural networks have been proposed for prediction [14]. Zhang et al. [1] presented a novel system that automatically parses streamed console logs and detects

early warning signals for IT system failure prediction based on the LSTM neural network. Susto et al. [15] developed a multiple classifier machine learning methodology to deal with the unbalanced datasets that arise in maintenance classification problems. Baban et al. [16] used a fuzzy logic approach to develop a decision-making system that allows determining the lifetime of the needle and plane predictive maintenance of the needle of a sewing machine. However, it required expert knowledge and depends on datasets of small quantity.

The above works have performed well to detect or predict the univariate anomaly for the IoT applications. However, we cannot explain why and how these approach works. Besides, more and more IoT applications need to analyze and identify the root causes of the anomalies, while these approaches cannot answer the root causes.

Mining temporal dependency provide essential clues to identify the cause relationship among anomalies [17]. Several types of dependent pattern mining tasks have been induced from practical problems and carefully studied. Song and et al. mined activity dependencies (i.e., control dependency and data dependency) to discover process instances when event logs cannot meet the completeness criteria [7]. A dependency graph is utilized to mine process instances. However, the authors do not consider the dependency among events. Plantevit et al. presented a new approach to mine temporal dependencies between streams of interval-based events [8]. Friedberg et al. proposed a novel anomaly detection approach. It keeps track of system events, their dependencies and occurrences, and thus learns the normal system behavior over time and reports all actions that differ from the created system model [18].

In this paper, we introduce the temporal dependency into the predictive maintenance to improve the explanation of prediction approaches and discovery the root cause of anomaly.

6 Conclusion

In this paper, we try to propose an effective and explainable approach to predict the anomaly based on mining the temporal dependencies from multi-sensor event sequences. To reach this goal, we detect “univariate anomaly” events from sensor data and output multi-sensor event sequences. Then, we transform the temporal dependency mining problem into a frequent co-occurrence pattern mining problem. Furthermore, a lot of experiments have been done to show the effectiveness of our approach based on a real dataset from a coal power plant. But the speed of our method can also be improved. So, for future work, we are interested in parallel optimization algorithms to speed up our method.

Acknowledgement. This work is supported by “The National Key Research and Development Plan (No: 2017YFC0804406), Public Safety Risk Prevention and Control and Emergency Technical equipment”, and the “Key Project of the National Natural Science Foundation of China No. 61832004 (Research on Big Service Theory and Methods in Big Data Environment)”.

References

1. Zhang, K., Xu, J., Min, M.R.: Automated IT system failure prediction: a deep learning approach. In: International Conference on Big Data, pp. 1291–1300 (2016)
2. Sipos, R., Fradkin, D., Moerchen, F.: Log-based predictive maintenance. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1867–1876. ACM, New York (2014)
3. Song, F., Zhou, B., Sun, Q.: Anomaly detection and explanation discovery on event streams. In: Business Intelligence for the Real-Time Enterprises (2018)
4. Ishimtsev, V., Bernstein, A., Burnaev, E.: Conformal k-NN anomaly detector for univariate data streams. arXiv: Machine Learning, pp. 213–227 (2017)
5. Siffer, A., Fouque, A., Termier, A.: Anomaly detection in streams with extreme value theory. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Canada, pp. 1067–1075. ACM (2017)
6. Qiu, H., Liu, Y., Subrahmanya, N.A., Li, W.: Granger causality for time-series anomaly detection. In: Proceedings of the IEEE International Conference on Data Mining, ICDM 2012, pp. 1074–1079 (2012)
7. Song, W., Jacobsen, H.A., Ye, C., Ma, X.: Process discovery from dependence-complete event logs. IEEE Trans. Serv. Comput. **9**(5), 714–727 (2016)
8. Plantevit, M., Robardet, C., Scuturici, V.M.: Graph dependency construction based on interval-event dependencies detection in data streams. Intell. Data Anal. **20**(2), 223–256 (2016)
9. Yan, Y., Luh, P.B., Pattipati, K.R.: Fault diagnosis of HVAC air-handling systems considering fault propagation impacts among components. IEEE Trans. Autom. Sci. Eng. **14**(2), 705–717 (2017)
10. Cao, Y., Liu, C., Han, Y.: A frequent sequential pattern based approach for discovering event correlations. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 48–59. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_5
11. Zeng, C., Tang, L., Zhou, W.: An integrated framework for mining temporal logs from fluctuating events. IEEE Trans. Serv. Comput. **12**(2), 199–233 (2019)
12. Domingues, R., Filippone, M., Michiardi, P., Zouaoui, J.: A comparative evaluation of outlier detection algorithms: experiments and analyses. Pattern Recogn. **74**, 406–421 (2018)
13. Yagci, A.M., Aytekin, T., Gurgun, F.S.: Scalable and adaptive collaborative filtering by mining frequent item co-occurrences in a user feedback stream. Eng. Appl. Artif. Intell. **58**, 171–184 (2017)
14. Warriach, E.U., Tei, K.: Fault detection in wireless sensor networks: a machine learning approach. In: 16th IEEE International Conference on Computational Science and Engineering, CSE 2013, Sydney, NSW, pp. 758–765 (2013)
15. Susto, G.A., Schirru, A., Pampuri, S.: Machine learning for predictive maintenance: a multiple classifier approach. IEEE Trans. Ind. Inform. **11**(3), 812–820 (2015)
16. Baban, C.F., Baban, M., Suteu, M.D.: Using a fuzzy logic approach for the predictive maintenance of textile machines. J. Intell. Fuzzy Syst. **30**(2), 999–1006 (2016)
17. Perng, C., Thoenen, D., Grabarnik, G., Ma, S., Hellerstein, J.: Data-driven validation, completion and construction of event relationship networks. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 729–734. ACM (2003)
18. Friedberg, I., Skopik, F., Settanni, G., Fiedler, R.: Combating advanced persistent threats: from network event correlation to incident detection. Comput. Secur. **48**, 35–57 (2015)