Finding Anomalies in SCADA Logs Using Rare Sequential Pattern Mining

Anisur Rahman[1], Yue Xu[2], Kenneth Radke[2], and Ernest Foo[2]

Queensland University of Technology, Brisbane, Australia
[1]{anisur.rahman}@hdr.qut.edu.au
[2]{yue.xu,k.radke,e.foo}@qut.edu.au

Abstract. Pattern mining is a branch of data mining used to discover hidden patterns or correlations among data. We use rare sequential pattern mining to find anomalies in critical infrastructure control networks such as supervisory control and data acquisition (SCADA) networks. As anomalous events occur rarely in a system and SCADA systems' topology and actions do not change often, we argue that some anomalies can be detected using rare sequential pattern mining. This anomaly detection would be useful for intrusion detection or erroneous behaviour of a system. Although research into rare itemsets mining previously exists, neither research into rare sequential pattern mining nor its applicability to SCADA system anomaly detection has previously been completed. Moreover, since there is no consideration to events order, the applicability to intrusion detection in SCADA is minimal. By ensuring the events' order is maintained, in this paper, we propose a novel Rare Sequential Pattern Mining (RSPM) technique which is a useful anomaly detection system for SCADA. We compared our algorithm with a rare itemset mining algorithm and found some anomalous events in SCADA logs.

Keywords: frequent pattern, rare pattern, SCADA, generator pattern.

1 Introduction

Anomaly detection is one step of several safeguarding measures applied in critical infrastructure (CI) control networks, such as supervisory control and data acquisition (SCADA). The SCADA system is used to monitor and control the CIs from a remote location. SCADA systems are interlinked with each other, so attacks on SCADA can cause devastating impacts to other dependent infrastructures, environments and even to human lives [1]. Many SCADA systems use conventional IT technology as a backbone to communicate with field devices, so they are prone to be attacked using standared IT network vulnerabilities. Anomaly detection for SCADA systems is important and challenging because of the constant changes in attack patterns. Therefore, it is almost impossible to keep the system protected from increasingly diversified attacks [2].

SCADA are distinguished from traditional IT networks because the normal or regular behaviour of SCADA system can be predicted using frequent sequential pattern or regular system behaviour. However, rare pattern or irregular behavior of this system which deviates from normal behaviour could be considered as anomalous events. Therefore, we argue that rare or infrequent sequence of actions can be considered as an anomalous event, and if analyzed we can find the cause could be either cyber-attacks, system failures, or later inclusion of a benign or novel event. In this paper, we propose a novel Rare Sequential Pattern Mining (RSPM) method for anomaly detection from SCADA logs.

2 Related Work

There has been several research in finding anomalies in SCADA systems from diverse perspectives. Some works are at the communication protocol level [3]; however, only a single work uses SCADA logs where Hadžiosmanovič et al. [4] used itemset mining for threat identification. Manganaris et al. [5] show that the absence of frequent events or set of events can be considered as an anomaly. Clifton et al. [6] applied data mining technique to identify the normal behavior of a system based on frequent occurrence of an alarm event and later filtered them out from suspicious events lists. Barbara et al. [7] defined users normal behavior using data mining association rules from network traffic data to train a model. Later, in their model they looked for any deviation in association rules and considered as abnormal or anomalous behavior of the system and users.

So, there have been some works in finding rare or infrequent itemset mining. However, these works can not be used to find anomalies in SCADA systems as they do not preserve itemsets' order [8, 4]. To the best of our knowledge, until now there has been no work in rare sequential pattern mining for anomaly detection.We are motivated by the work of Szathmary et al. [8] where the authors used minimal frequent itemset generators to find rare patterns. However, their method cannot find correlation among the events. We apply almost similar idea, but instead of itemset we use sequence which preserves the events' order of occurence that results in correlation among the events. Therefore, in this paper, we use rare sequential pattern minig (RSPM) which is a branch of sequential pattern mining first introduced by Agrawal and Srikant [9].

3 Proposed Method

To define our problem, we introduce some related theories from [10] which are applied in our RSPM algorithm, and in other sections of this paper.

Definition 1 (Sequence): *Let* $I = \{i_1, i_2, ..., i_l\}$ *be a set of all items. An itemset* $I_x = \{i_1, i_2, ..., i_m\} \subseteq I$ *is a nonempty and unordered set of distinct items. A sequence s is an ordered list of itemsets or events denoted as* $\langle I_1, I_2, I_3, ..., I_n\rangle$ *such that* $I_k \subseteq I$ $(1 \leq k \leq n)$.

Definition 2 (Sequential Database SDB): *A sequential database SDB is a set of sequences, i.e.,* $SDB = \{s_1, s_2, s_3, ..., s_p\}$*, where* s_j *is a sequence,* $1 \leq j \leq p$.

For example, Table 1 shown below is an example of a sequential database SDB containing four sequences. The first sequence SID1 is composed of 5 itemsets. The first itemset is {1} which is followed by itemset {1, 2, 3}. For application domains, the items in one itemset are often considered occur at the same time.

Definition 3 (Sequence containment): *A sequence* $S_a = \langle A_1, A_2, ..., A_n \rangle$ *is said to be contained in a sequence* $S_b = \langle B_1, B_2, ..., B_m \rangle$ *if and only if there exist integers* $1 \leq i_1 \leq i_2 \leq ... \leq i_n \leq m$, *such that,* $A_1 \subseteq B_{i1}$, $A2 \subseteq B_{i2}$, ..., $A_n \subseteq B_{in}$ *and this is denoted as* $S_a \sqsubseteq S_b$. *In this case* S_a *is considered as a sub-pattern of* S_b *and* S_b *is also said to be super-pattern of* S_a.
For example, in Table 1 sequence $\langle\{5\}, \{1, 6\}, \{2\}\rangle$ is contained in sequence SID4.

Definition 4 (Support): *The support of a sequential pattern* S_a *in a sequential database SDB is determined by the number of sequences* $S \in$ *SDB, such that,* $S_a \sqsubseteq S$ *and it is denoted by supSDB*(S_a).For example, the pattern $\langle\{1, 2\}, \{6\}\rangle$ is found in 2 sequences in Table 1 and hence the support is 2.

Definition 5 (Frequent Sequential Pattern): *Let minsup be a user-defined threshold and SDB is a sequential database. A sequence S (also called a sequential pattern) is considered frequent if and only if supSDB (S)* $\geq$ *minsup.*

Definition 6 (Sequential Generator): *A sequential pattern* S_a *is said to be a generator if there is no other sequential pattern* S_b *such that* $S_b \sqsubseteq S_a$ *and their supports are equal.* For example, in the sequence database SDB given in Table 1 both $\langle\{5\}, \{2\}\rangle$ and $\langle\{6\}, \{3\}\rangle$ are Generator Sequential Patterns.

Table 1. A sequential database SDB

Sequence ID	Sequences
SID1	$\langle\{1\}, \{1, 2, 3\}, \{1, 3\}, \{4\}, \{3, 6\}\rangle$
SID2	$\langle\{1, 4\}, \{3\}, \{2, 3\}, \{1, 5\}\rangle$
SID3	$\langle\{5, 6\}, \{1, 2\}, \{4, 6\}, \{3\}, \{2\}\rangle$
SID4	$\langle\{5\}, \{7\}, \{1\ 6\}, \{3\}, \{2\}, \{3\}\rangle$

3.1 Description of RSPM Algorithm

In rare sequential pattern mining, the sequences that fail to meet the *minsup* are known as rare sequences. For example, if the user defined *minsup* is 2 then the sequential patterns $\langle\{7\}, \{1\}\rangle$ and $\langle\{1\}, \{5\}\rangle$ are found to be rare since they fall below the *minsup.* The basic idea is to form a new sequential pattern by combining two minimal generators and the infrequent combinations are considered rare patterns. We are motivated by the work presented in [8] that finds rare itemsets based on minimal generators. However, we propose to find rare sequential patterns based on sequential generators because they are the smallest or minimal patterns of an equivalent class. Usually, shorter patterns are frequent, while by nature, longer patterns are likely to be infrequent or rare and their combination can be even more infrequent. Therefore, it is likely that a combination of two minimal generators would be rare or infrequent, and this rare sequential pattern can be considered interesting and deserves further investigation. Given two sequential patterns, there could exist different ways to combine them to form a new sequential pattern. For example, for the two frequent sequential patterns s_1 = $\langle\{4\}, \{3\}, \{2\}\rangle$ and $s_2 = \langle\{1, 2\}, \{6\}\rangle$ generated from the dataset in Table 1, at the sequence level, we can combine them in two different orders $\langle s_1, s_2\rangle$ and $\langle s_2, s_1\rangle$. For example, $\langle\{4\}, \{3\}, \{2\}, \{1, 2\}, \{6\}\rangle$ and $\langle\{1, 2\}, \{6\}, \{4\}, \{3\}, \{2\}\rangle$. In this case, in the resulting sequence, each of the two original sequences is intact. The order of itemsets from one original sequence is preserved. However, if we only want to preserve the original order of itemsets and do not require the integrity of the original sequences, we could have sequences like $\langle\{4\}, \{1, 2\}, \{3\}, \{2\}, \{6\}\rangle$, $\langle\{1, 2\}, \{4\}, \{3\}, \{6\}, \{2\}\rangle$, and much more. So, we preserved both the integrity and itemset order of the original sequence.

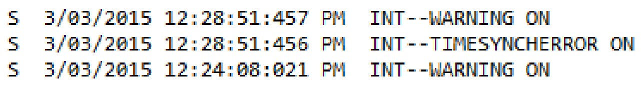
The inputs of this algorithm (shown below) are a sequence database SDB and a user defined threshold value as *minsup*. This algorithm will produce a list of minimal rare sequential patterns. At the beginning, minimal sequential generator patterns (mSGP) and frequent sequential patterns are generated from the sequence database SDB in steps 3 and 4. Then for each pair of generators' combinations are checked against the frequent sequential patterns (FSP) as described in step 6 to step 12.

Algorithm 1: Rare Sequential Pattern Mining Algorithm

```
1: Input:   SDB, minsup // A sequential database and minimum support
2: Output: RSP // A set of rare sequential patterns
3: G := $\langle g_1, g_2, ...\rangle$ // a list of minimal sequential generators
4: FSP := $\langle s_1, s_2, ...\rangle$ // a set of frequent sequenial patterns
5: RSP := { }
6: for $g_i$ in G do
7:     for $g_j$ in G do
8:         if $\langle g_i, g_j\rangle \notin$ FSP and $\exists$s $\in$ SDB and $\langle g_i, g_j\rangle \sqsubseteq s$ then
9:             RSP := RSP $\cup$ {$\langle g_i, g_j\rangle$}
10:        else
11:            if $\langle g_j, g_i\rangle \notin$ FSP and $\exists$s $\in$ SDB and $\langle g_j, g_i\rangle \sqsubseteq s$ then
12:                RSP := RSP $\cup$ {$\langle g_j, g_i\rangle$}
13:            end if
14:        end if
15:    end for
16:    Return RSP
17: end for
```

4 Experimental Setup

We assume that attackers cannot alter or delete the SCADA logs that we use as our datasets. We use four real SCADA logs Datasets. First dataset (Dataset-1; shown in Figure 1) was collected from the logs on an Intelligent Electronic Device that controls an electrical substation while the second, third, and fourth dataset (Dataset-2, Dataset-3, and Dataset-4 respectively) were collected from our three different SCADA laboratory setup. In case of Dataset-1, Log data includes recorded events on that substation which were recorded only when there occurred any system errors.



```
S  3/03/2015 12:28:51:457 PM  INT--WARNING ON
S  3/03/2015 12:28:51:456 PM  INT--TIMESYNCHERROR ON
S  3/03/2015 12:24:08:021 PM  INT--WARNING ON
```

Fig. 1. A partial view of Dataset-1

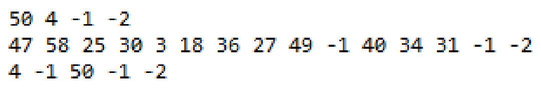
On the contrary, The Dataset-2 comes from a water tank system in our laboratory, which consists of two water tanks and a pump that moves water from a lower tank into the upper tank. Gravity allows water in the upper tank to move back into the lower tank. Dataset-3 collected from a compressed air pipeline or reactor system. An air compressor pumps air into the pipe system and increases the air pressure. At a given value the air pressure is released. Once the air is released, the compressor starts up again building pressure in the pipe system. Finally, Dataset-4 is a conveyer system that moves objects along a conveyer belt. Dark and light objects are separated into Left and Right directions before returning to the beginning of the system. It is possible to sort objects in opposite directions.

In a particular day, there was a training session in our SCADA lab from 9.30am to 4.00pm. Three system devices (Tank, Reactor, and Conveyer) were switched on and started functioning smoothly. However, the system was compromised in the later half of the day, and all events were recorded.

4.1 Data preprocessing

In data preprocessing steps, the raw logs from all datasets (Dataset-1 to Dataset-4) have been cleaned and the necessary informative features were selected. It is to be noted that in the log entries of the electrical substation (from where Dataset-1 created), it has been observed that during two minutes time duration the system performs a series of sequential events to bring the system to normal state from erroneous state. Therefore, these sequence of events have been identified as a single sequence which build the sequential database SDB-1 (shown in Figure 2).

Here, the numbers represent individual events or itemsets of raw logs (Dataset-1) while "-1 "indicates ending of events or itemsets and consecutive "-1 -2 "signals



```
50 4 -1 -2
47 58 25 30 3 18 36 27 49 -1 40 34 31 -1 -2
4 -1 50 -1 -2
```

Fig. 2. A partial view of SDB-1

the end of a sequence. However, as in Dataset-2 through Dataset-4, events are recorded in every second, these events are considered as a single item in the sequence which build the databases SDB-2 through SDB-4 respectively.

4.2 Experiment and Results

We performed our experiment with python programming and SPMF [11] tool, which is an open source framework for sequential pattern mining. At first, we generate frequent sequential patterns (FSP) and minimal sequential generator patterns (mSGP). Later, the combination of mSGP are compared with FSP to prune frequent patterns. The remaining rare patterns are once again compared with the sequence database SDB. These patterns are once again found rare and considered anomalous events. However, the rare patterns which are not found in SDB are considered as non-present patterns.

Table 2. A partial view of results for Dataset-3.

Patterns by RSPM	Patterns by Rare Itemset
⟨{19}, {57}⟩	{19, 57}
⟨{58}, {19}⟩	***{19, 58}***
⟨{100}, {74}⟩	***{74, 100}***

We also applied Szathmary et al.'s rare itemset mining algorithm with the same datasets (Dataset-1 through Dataset-4) used in our RSPM algorithm for comparison. For example, their algorithm has identified rare itemset pattern {*19, 58*} against our RSPM algorithm's rare sequential pattern ⟨{*58*}, {*19*}⟩ (shown in Table 2); However, we checked with the original log sequence events and found that the events occurrence order is 58 followed by 19 and not in events' reverse order. We argue that a particular sequential events' order can lead to a particular result and it is very important and significant in our experiment. For example, the following sequential ordered events are a regular system profile for filling a tank reservoir:

1. Turn on pump.
2. Wait for water level to reach 40%.
3. Turn off pump.

If the system runs with the above events mentioned in the order, then the tank pump turns off after the water level reaches to the 40% of its capacity label.

However, if the above events are performed in a different order as mentioned below:

1. Turn off pump.
2. Wait for water level to reach 40%.
3. Turn on pump.

then the system floods; therefore, we can say that only the rare patterns cannot be effective in identifying intrusion rather we need rare patterns with ordered events for effective intrusion detection into a system.

5 Discussion

The experimental results for all datasets (Dataset-1 to Dataset-4) show that our RSPM algorithm found not only the rare patterns as identified by Szathmary et al.'s algorithm but also keeps the events occurrence order, which their algorithm did not consider. For example, rare pattern $\langle\{14\}, \{7\}, \{10\}\rangle$ from Dataset-4 has been identified by our algorithm; however, Szathmary et al.'s algorithm identified this pattern as $\{7, 10, 14\}$ even though these are two different patterns considered in sequential pattern domain. Here, the numbers in the pattern $\langle\{14\}, \{7\}, \{10\}\rangle$ represents SCADA log sequence events ⟨{Conv_Read_Conv_HMI_Direction(5)_0}, {Conv_Read_Conv_Present_PE(5)_0}, {Conv_Run_Status(5)_0}⟩ in Dataset-4. We traced this rare pattern in the original log dataset. However, we did not find pattern $\{7, 10, 14\}$ identified by their algorithm in the original log sequence as they did not preserve the sequence order. Different ordered sequential events' can produce different end results. Therefore, our rare sequential pattern can be effective in finding the correlation between consequences and actions.

Moreover, we found the original sequence as a frequent sequence which ends with the event "Conv_Run_Status(5)_-1"; however, in the rare sequence pattern the sequence ends with "Conv_Run_Status(5)_0"which is a deviation from the regular profile of the system. Later, we traced back this deviation in the log file and found that during the events' time period the converyer belt direction was reversed although it was supposed to be moving in other direction. This abnormal incident occured in the second part of the training day when the system was compromised. Therefore, we came to the conclusion that this rare sequence was an anomalous event which happened due to system compromise.

Similarly, we traced back rare sequential patterns $\langle\{19\}, \{57\}\rangle$, $\langle\{58\}, \{19\}\rangle$, and $\langle\{100\}, \{74\}\rangle$ from Dataset-3 and $\langle\{29\}, \{50\}\rangle$ from Dataset-2. In all cases, we found that these rare sequences should not happen in the logs during the specified time period. Therefore, we also believe that these are anomalous events. However, as we do not have a complete labeled test dataset from log files, we cannot find the ratio of false positive and negative. But, to find whether our algorithm can detect anomalous events, we have manually rearranged the order of some sequences with Dataset-1, and our algorithm detects these changes as rare sequences.

6 Conclusion and Future Work

In this paper we have presented RSPM, a novel approach for anomaly detection from SCADA logs using rare sequential pattern mining. However, it may be possible for the adversaries provided that they repeat the malicious events multiple times to evade this technique. In future, we will extend this work to generate all (from minimal to maximal) rare sequential patterns to test which patterns become more effective in detecting intrusion. We will also validate and find computational performances of our methodology with large volume of publicly available labeled SCADA logs. Moreover, we will compare our algorithm with other works as to anomaly detection using non sequential pattern outside SCADA or CIs.

References

1. P. Pederson, D. Dudenhoeffer, S. Hartley, and M. Permann, "Critical infrastructure interdependency modeling: a survey of US and international research," *Idaho National Laboratory*, vol. 25, p. 27, 2006.
2. M. Cheminod, L. Durante, and A. Valenzano, "Review of Security Issues in Industrial Networks," *IEEE Trans. on Ind. Informat*, vol. 9, no. 1, pp. 277–293, 2013.
3. S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. of the SCADA Security Scientific Symp.*, vol. 46, 2007, pp. 1–12.
4. Hadžiosmanovič, Dina and Bolzoni, Damiano and Hartel, Pieter H, "A log mining approach for process monitoring in SCADA," *Int. J. of Inform. Security*, vol. 11, no. 4, pp. 231–251, 2012.
5. S. Manganaris, M. Christensen, D. Zerkle, and K. Hermiz, "A data mining analysis of RTID alarms," *Computer Networks*, vol. 34, no. 4, pp. 571–577, 2000.
6. C. Clifton and G. Gengo, "Developing custom intrusion detection filters using data mining," in *IEEE Proc. 21st Century Military Commun.*, vol. 1, 2000, pp. 440–443.
7. D. Barbara, N. Wu, and S. Jajodia, "Detecting Novel Network Intrusions Using Bayes Estimators," in *1st SIAM Conf. on Data Mining*, 2001, pp. 1–17.
8. L. Szathmary, A. Napoli, and P. Valtchev, "Towards rare itemset mining," in *19th IEEE Int. Conf. on Tools with Artificial Intell.(ICTAI 2007)*, vol. 1, 2007, pp. 305–312.
9. R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. of the 11th Int. Conf. on Data Engineering.* IEEE, 1995, pp. 3–14.
10. P. Fournier-Viger, A. Gomariz, M. Šebek, and M. Hlosta, *VGEN: Fast Vertical Mining of Sequential Generator Patterns.* Springer, 2014, pp. 476–488.
11. P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C. Wu., and V. S. Tseng, "SPMF: a Java Open-Source Pattern Mining Library," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 3389–3393, 2014.