ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 20-04-13

j.inst.Korean.electr.electron.eng.Vol.24,No.4,1011~1016,December 2020
http://dx.doi.org/10.7471/ikeee.2020.24.4.1011


# Detecting Anomalies in Time-Series Data using Unsupervised Learning and Analysis on Infrequent Signatures

Xingchao Bian*★

## Abstract

We propose a framework called Stacked Gated Recurrent Unit - Infrequent Residual Analysis (SG-IRA) that detects anomalies in time-series data that can be trained on streams of raw sensor data without any pre-labeled dataset. To enable such unsupervised learning, SG-IRA includes an estimation model that uses a stacked Gated Recurrent Unit (GRU) structure and an analysis method that detects anomalies based on the difference between the estimated value and the actual measurement (residual). SG-IRA's residual analysis method dynamically adapts the detection threshold from the population using frequency analysis, unlike the baseline model that relies on a constant threshold. In this paper, SG-IRA is evaluated using the industrial control systems (ICS) datasets. SG-IRA improves the detection performance (F1 score) by 5.9% compared to the baseline model.

*Key words : Industrial Control System Security Threat Detection, Anomaly Detection, Time-series data, Stackd-GRU, Frequency Analysis, Unsupervised learning, TaPR.*

## Ⅰ. Introduction

Recently, cyber security threats to the control systems of national infrastructure and industrial facilities have continued to increase. Countries around the world are committed to developing security technologies in response to cyber attacks that can cause irreparable damage to countries and societies on vital national facilities. In particular, the dataset that can accurately reflect the characteristics of the field control system and contains various types of cyber attacks of the control system.

Deep neural networks such as Long-short Term Memory (LSTM) or Gated Recurrent Unit (GRU) demonstrated its effectiveness on time-series data [1]. However, training of such models requires a large amount of training data labeled by humans or collected from the previous attack incidents. Although the consequences of cyber attacks on industrial system are critical once succeeded, it is difficult to collect a large number of attack samples before the attack is actually attempted is difficult. Therefore, to construct a

* Department of Software, College of Computing, Sungkyunkwan University
★ Corresponding author
E-mail : bxc8802@g.skku.edu, Tel : +82-31-290-7698
※ Acknowledgment
This work was supported by Institute for Information & communication Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2019-0-01343, Regional strategic industry convergence security core talent training business) and (No.2019-0-00533, Research on CPU vulnerability detection and validation).
Manuscript received Nov. 26, 2020; revised Dec. 13, 2020; accepted Dec. 19, 2020.

machine learning model that effectively deters an attack attempt must be based on unsurpervised learning that learns is able to distinguish abnormal system behavior from normal states.

In this paper, we present an unsupervised learning framework called Stacked Gated Recurrent Unit-Infrequent Residual Analysis (SG-IRA). SG-IRA's detection method is improved upon the HAI 2.0 baseline model from Industrial Control System Security Threat Detection AI Competition [2], which was conducted in the DACON data competition platform.

SG-IRA mainly consists of an estimation model that predicts the sensor measurements of the next time-slot, and an analysis method that detects the anomalies (*i.e.*, possible attacks) based on the differences between the estimated values and the actual measurement (residual). SG-IRA's analysis method dynamically finds the detection threshold through the frequency analysis using the observed statistics during the detection phase.

SG-IRA improves the detection performance (F1 score) by 5.9% on HAI 2.0 validation dataset compared to the baseline model.

## II. Backgrounds

### 1. HAI dataset

The HIL-based augmented ICS security (HAI) dataset is the first cyber-physical system (CPS) dataset that was collected on the HAI testbed [2]. The HAI testbed comprises three physical control systems, namely GE turbine, Emerson boiler, and FESTO water treatment systems, combined through a dSPACE hardware-in-the-loop (HIL) simulator [3].

The dataset has multiple channels of measurements (e.g., sensors, actuators, control devices), that represent the current status of the system, and one measurement is obtained every second.

The training dataset was collected with no attacks while the test or validation dataset contains simulated attacks. That is, the model has to learn a detection mechanism without seeing any example attack data.

The HAI dataset has the following two sets of datasets from two different target configurations.

(1) HAI dataset 1.0
  a. 59 measurement channels
  b. Training dataset: 550,800 seconds
  c. Test dataset: 444,600 seconds
  d. The test dataset includes 38 attacks combining 14 attack primitives [3].

(2) HAI dataset 2.0
  a. 79 measurement channels
  b. Training dataset: 921,603 seconds
  c. Validation dataset[1]: 43,201 seconds
  d. The validation dataset includes 5 attacks.

Table 1. Partial training dataset in HAI dataset 2.0.

| | Time | C01 | C02 | C03 | C04 | C05 | ... | C79 |
|---|---|---|---|---|---|---|---|---|
| 0 | 2020-07-11 00:00:00 | 395.19528 | 12 | 10 | 52.80456 | -1.2648 | ... | 6.0951 |
| 1 | 2020-07-11 00:00:01 | 395.14420 | 12 | 10 | 52.78931 | -1.3147 | ... | 5.9262 |
| 2 | 2020-07-11 00:00:02 | 395.14420 | 12 | 10 | 52.79694 | -1.4032 | ... | 5.8101 |
| 3 | 2020-07-11 00:00:03 | 395.19528 | 12 | 10 | 52.79694 | -1.6074 | ... | 5.7509 |
| 4 | 2020-07-11 00:00:04 | 395.34866 | 12 | 10 | 52.79694 | -1.7811 | ... | 5.8547 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 921601 | 2020-08-10 10:59:59 | 387.73221 | 12 | 10 | 66.72057 | -1.4912 | ... | 6.4150 |
| 921602 | 2020-08-10 11:00:00 | 387.52774 | 12 | 10 | 66.72057 | -1.5727 | ... | 6.6288 |

### 2. Time-Series Aware Precision and Recall

TaPR is time-series aware precision and recall,

1) For the HAI dataset 2.0, we used the validation dataset to evaluate SG-IRA since the DACON challenge did not disclose the test dataset at the time of the competition.

which are appropriate for evaluating anomaly detection methods in time-series data[4]. As shown in Fig. 1, the goal is to detect the scope affected by attacks such as $\alpha_1$ and $\alpha_2$ scope.

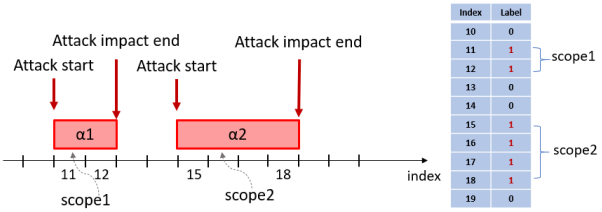


Fig. 1. Detect the scopes affected by attacks.

The variety of the detected anomalies is more important based on two scoring strategies in TaPR metrics. The two strategies are that detection scoring which is called TaR (i.e., how many anomalies are detected) and portion scoring which is called TaP (i.e., how precisely each anomaly is detected). In addition, TaPR metrics give lower scores to those instances which labeled as normal although they are affected by their precedent anomaly as they're probably anomalous.

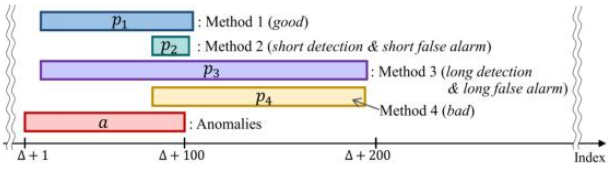


Fig. 2. TaPR evaluation metrics.

Fig. 2 indicates detection result $P_2$ is better than instance $P_3$ and $P_4$ for attack $\alpha$ as the more accurate the higher score, the more inaccurate the lower score.

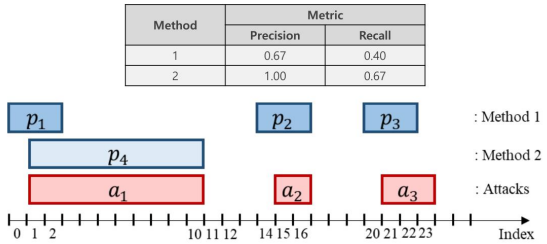


Fig. 3. Precision and Recall are unfit to assess whether various attacks have been detected.

And the reason why don't use Precision and Recall is that they get very high score when $\alpha 1$ is only detected as shown in Fig. 3.

※ Fig. 1-3 is referred or modified from the eTaPR[5] which the copyright belongs to the author.

### 3. System Structure

Figure 4 describes the overall structure of the SG-IRA platform. The framework contains a simple data preprocessing unit, an estimation model that uses a stacked-GRU, and the anomaly detection mechanism through the frequency analysis.

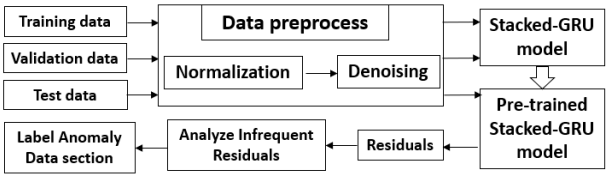


Fig. 4. Overall scheme of the proposed time-series anomaly data detection framework.

### 4. Data preprocessing

The input to the detection system is raw sensor measurements that have a wide variety of ranges depending on the measurement channel and contain noisy measurements. Therefore, SG-IRA normalizes the input data and performs noise reduction using an exponential weighted function.

## III. SG-IRA Mechanism

### 1. Motivation for Frequency Analysis

In [6], the authors propose a detection mechanism for time series multi-scale anomaly based on the Haar wavelet transform. This mechanism categorizes the measurements by computing the slope and length of two neighbor sampling points. The number of observations in each category bucket (*Support Count*) becomes the basis of selecting the detection thresholds.

The frequency analysis in SG-IRA is based on the residual value, which is defined as the difference

between the predicted value generated by the estimation model and the real measurement.

## 2. Estimation Model

Our estimation model is based on a stacked-GRU structure. The input to the model is the measurement in the sliding window that takes a portion of the time-series data. In our evaluation, the length of the window is set to 89 seconds (*i.e.*, 89 data points). The output from the model estimates the value after the sliding window (*i.e.*, the expected value at the 90-th second), and the residual (the difference between the estimation and actual measurement) is obtained.

The basic assumption of the detection mechanism is that the larger the residual is, the higher the possibility the observation is an attack. Since we trained the estimation model on attack-free situations only, a well-trained estimation model would closely predict the normal behaviors. A large residual value indicates an unfamiliar measurement that was not included in the training set, and it is an indication of a possible attack.

The estimation model uses a 3-layer bidirectional GRU. The size of the hidden cell is set to 100 and dropout is not used. The model uses a skip connection to export the first value of the window plus the output of the RNN. The loss function is mean squared error (MSE) and the AdamW optimizer is used.

## 3. Infrequent Residual analysis

The detection mechanism needs a threshold value to distinguish an attack based on the scale of the residual value. In the baseline model provided by DACON, a static threshold value is adopted to distinguish an attack from normal situations. However, it is challenging to choose an appropriate threshold because the detection quality is largely determined by the threshold level. A larger threshold may fail to detect some attacks, whereas a smaller threshold may tend to flag many benign signatures as anomalies.

SG-IRA adopts a heuristic mechanism that automatically learns an appropriate detection threshold using the frequency analysis similar to the mechanism discussed in [6].

The mechanism operates in the following steps:

1) We aggregate the residual value across $n$ channels (e.g., $n$ is 59 and 79 for HAI 1.0 and 2.0, respectively) into "*anomaly scores*," *short for* $AS_t$ using their average

$$AS_t = \frac{C1_t + C2_t + C3_t + ... + Cn_t}{n} \quad (1)$$

where $t$ represents the time-slot. Optionally, we may use a sliding window over a range of residual values if the frequency of the measurement is high (*i.e.*, $t$ become a range of time-slots, instead of indicating a specific second).

2) We categorize the observed anomaly scores into category buckets

$$AS_j \in [AS_t, AS_t + W_b) \quad (2)$$

where the width of a bucket ($W_b$) is a parameter of SG-IRA.

3) We count the number of anomaly scores categorized into each bucket (*Support Count*) along with the entire dataset. The Support Count of an anomaly score obtained from an infrequent situation would be lower than other anomaly scores.

4) We rank the category buckets in the ascending order of their Support Count and use the $k$-th bucket in the sorted order to determine the detection threshold. We set the smallest value in the range of the $k$-th bucket (*i.e.*, ASj of bucket $j$, where bucket $j$ is the k-th smallest Support Count) as *minAS*, If the observed anomaly score is larger than minAS, the measurement is classified as an anomaly.

## 4. Merging Intermittent Attack Signatures

An attack typically happens over a continuous time period per incident. Therefore, an anomaly detection system needs to indicate a continuous-

time as a single anomaly incident to match the duration of the attack event. The TaPR metric in II. 2 considers such continuity in the evaluation instead of just comparing the number of detected events. If there are two adjacent time periods where both of them are classified as anomalies, it is highly likely that they have resulted from a single continuous attack attempt. Therefore, SG-IRA considers those two periods as a continuous attack if the interval between two anomaly periods is less than 500 seconds. SG-IRA performs post-processing to bridge the gap between the adjacent anomaly periods.

We empirically selected the gap limit as 500 seconds based on the validation dataset since the minimum interval of two separate attack attempts is greater than 2,000 seconds in the dataset. Although we set the gap limit to 500 seconds to ensure we can cover various attack patterns, the actual time slots connected by is usually one or two seconds.

## IV. Results

SG-IRA has three hyper-parameters: $st$ is the time-step stride of the estimation model, $W_b$ is the width of the bucket in the frequency analysis, and $k$ is the rank of the bucket selected for the detection threshold. The baseline has two hyper-parameters: $st$ is the time-step stride of the estimation model and $th$ is the static threshold.

At first, the time-step stride of the estimation model is set to 10, which is the same as the baseline model, and we obtained a similar or a little better evaluation as shown in experiments No. 2 and 5 in Table 2. Then, we modified the time-step from 10 to 1, which means that the estimation happens in a finer granularity (1 second), but the data prediction performs better as shown in experiments No. 1 and 4 in Table 2. As a result, we achieved a better F1 score and detection results than the baseline model. Specifically, we improved the F1-score performance by 5.9% on the HAI 2.0 validation dataset.

Table 2. TaPR Evaluation Experiments.

| No. | Model | Dataset | Data | Para. | F1 | TaP | TaR |
|---|---|---|---|---|---|---|---|
| 1 | SG-IRA | HAI1.0 | Test | $st$=1, $W_b$=0.01, $k$=5 | **0.811** | **0.934** | **0.711** |
| 2 | SG-IRA | | | $st$=10, $W_b$=0.01, $k$=5 | **0.808** | **0.897** | **0.735** |
| 3 | Baseline | | | $st$=10, $th$=0.1 | 0.792 | 0.924 | 0.693 |
| 4 | SG-IRA | HAI2.0 | Val. | $st$=1, $W_b$=0.1, $k$=6 | **0.977** | **0.968** | **0.805** |
| 5 | SG-IRA | | | $st$=10, $W_b$=1, $k$=3 | 0.913 | 0.859 | 0.974 |
| 6 | Baseline | | | $st$=10, $th$=0.04 | 0.918 | 0.954 | 0.884 |

## V. Discussions

Figures 5 and 6 compare two example cases with high and low TaPR scores. In Fig. 6, SG-IRA misclassified some normal situations as anomalies (f point). These are the points where the estimation model failed to accurately estimate the sensor measurements and resulted in high residual values. To reduce these cases, a better estimation model has to be trained either by using a better DNN model or a larger number of datasets.

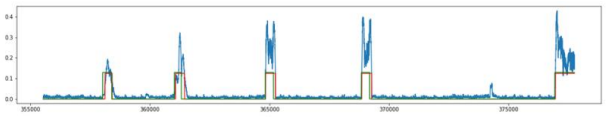

Fig. 5. An example of anomaly dectection on the HAI 1.0 test dataset. A case with a high F1 score is shown.

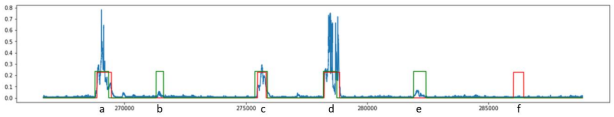


Fig. 6. An example of anomaly dectection on the HAI 1.0 test dataset. A case with a low F1 score is shown.

On the contrary, b and e points in Fig. 6 are the cases where SG-IRA failed to detect the attacks. In these cases, the residual value does not have a large difference compared to the

residual values from normal situations. Therefore, those cases are not caused by an inappropriate threshold value; those cases are caused by insufficient measurements, such as attacks through a hidden covert channel.

## VI. Conclusions

In this paper, we propose a time-series anomaly data detection framework called SG-IRA. The proposed method can be applied to detect anomaly data based on the dataset without pre-labeled attack samples. Compared to the previous baseline, which uses a static detection threshold, our mechanism is a heuristic mechanism that automatically learns an appropriate detection threshold.

## References

[1] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *In NIPS 2014 Workshop on Deep Learning,* 2014.
[2] DACON, Industrial Control Systems Security Threat Detection AI Competition https://dacon.io/ompetitions/official/235624/overview/
[3] H. Shin, W. Lee, J. H. Yun, H. Kim, "HAI 1.0: HIL-based Augmented ICS Security Dataset," *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, 2020.
[4] TaPR: W.-Hwang et al. "Time-Series Aware Precision and Recall for Anomaly Detection: Considering Variety of Detection Result and Addressing Ambiguous Labeling," *In Proc. of CIKM,* pp.2241-2244, 2019.
[5] eTaPR, https://www2.slideshare.net/daconist/etapr-237428659?ref=https://dacon.io/
[6] X. Chen, Y. Zhan, "Multi-scale anomaly detection algorithm based on infrequent pattern of time series," *Computational and Applied Mathematics,* VOL.214, pp.227-237, 2008.
DOI: 10.1016/j.cam.2007.02.027

## BIOGRAPHY

**Xingchao Bian** (Member)

2020 : BS degree in Computer Engineering, Inha University.
2020~present : MS degree in Software Engineering, Sungkyunkwan University.