2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)

Anomaly Detection in Cyber Physical Systems using Recurrent Neural Networks

Jonathan Goh, Sridhar Adepu, Marcus Tan and Lee Zi Shan
iTrust, Center for Research in Cyber Security,
Singapore University of Technology and Design
Singapore

***Abstract*—This paper presents a novel unsupervised approach to detect cyber attacks in Cyber-Physical Systems (CPS). We describe an unsupervised learning approach using a Recurrent Neural network which is a time series predictor as our model. We then use the Cumulative Sum method to identify anomalies in a replicate of a water treatment plant. The proposed method not only detects anomalies in the CPS but also identifies the sensor that was attacked. The experiments were performed on a complex dataset which is collected through a Secure Water Treatment Testbed (SWaT). Through the experiments, we show that the proposed technique is able to detect majority of the attacks designed by our research team with low false positive rates.**

***Keywords*—Anomaly detection, Cyber-physical systems, Recurrent neural network, Cumulative sum**

I. Introduction

Cyber-Physical Systems (CPSs) are interconnected physical systems for mission-critical tasks. Examples of such systems include water treatment and distribution plants, power grids and autonomous vehicles. Many of these systems are networked for remote monitoring and control. When such systems are connected to the internet, they become susceptible to cyber attacks. A few examples of cyber-attacks includes the 2016 cyber attack on a Ukraine power plant [9], the Stuxnet worm that targeted a nuclear power plant [5], and the insider threat on Australia's Moochy Water services that occurred in 2000 [14]. Hence, there is an impending need to secure such CPSs against such scenarios.

Behavioral-based approaches are one of the techniques used in intrusion detection systems. Such approaches are classified into supervised and unsupervised techniques. In the supervised approach ([4], [7], [8]), training labelled data that comprises of both normal and abnormal behaviours are provided to the model to learn. However, labelled data of CPSs are very difficult to obtain especially for attack data and also, simulated data may not be realistic. In such scenarios, unsupervised learning have the advantage of not requiring any abnormal data in the training phase. Although promising in detecting abnormality, majority of the work that utilises unsupervised learning ([12], [13]) has resulted in very high false positives.

In this work, we propose the use of a Long Short Term Memory Recurrent Neural Network (LSTM-RNN) to predict a sequence of data for anomaly detection. As anomalies or cyber-attacks typically occur over time, correlating time-series data provides us with information over time that can be used to better identify an anomaly. LSTM-RNN have demonstrated to be useful for learning sequences containing patterns of unknown length. In addition, stacking recurrent hidden layers in a neural networks has shown to capture the structure of the time series [10]. To the best of our knowledge, the only other works that utilises Recurrent Neural Networks in this domain are by Al-Jarrah et al. [3] and Malhotra et al. [11]. However, Al-Jarrah's work is in Intrusion Detection System (IDS) for network traffic monitor with the aim of detecting attacks and classifying them into host sweep or port scan. In [11], the authors used RNN for modelling several real-world time series data. Anomalies were detected based on the probability error above a pre-defined fixed threshold which causes many false positives. Ours differs by using LSTM-RNN for learning the temporal behaviour of the data in CPSs and using Cumulative Sum (CUSUM) for anomaly detection in this domain.

We use the LSTM-RNN as a predictor to model the normal behaviour and subsequently, the Cumulative Sum method to identify abnormal behaviours. This method is very useful in real-world CPS where instances of abnormal behaviours are rare. The goal of this paper is to provide a novel approach to behavioural-based intrusion detection in CPSs. The key contributions of this paper are:

1) Modelling of normal behaviour in a CPS using unsupervised deep learning through a data-driven approach
2) Identifying the sensor that exhibits abnormal behaviour
3) Validation of proposed method on Secure Water Treatment (SWaT) testbed[1]

The novelty of our work are as follows: our work is in the area of water critical infrastructure. More importantly, our prediction model is based on data obtained from a Secure Water Treatment (SWaT) testbed which is a scaled down replicate of a industrial water treatment plant, thus reflecting the complexity often found in a real plant. Next, our approach uses a time based neural network that takes into consideration a sequence of information as opposed to features obtained in a single second; and this allows for lower false positive rates. Lastly, not only does our proposed method detect anomalies, it is also capable of identifying which sensor the anomaly is occurring at.

The remaining of our paper is organised as follows: Section II describes our proposed method. In section III, we briefly

[1] http://itrust.sutd.edu.sg/research/testbeds/secure-water-treatment-swat/

1530-2059/17 $31.00 © 2017 IEEE
DOI 10.1109/HASE.2017.36

describe the data and the various attack scenarios. We present our results on the Secure Water Treatment (SWaT) dataset [6] in section IV. Section V concludes this paper.

II. Methodology

A. Formulation of Anomaly Detection Problem using LSTM

Traditional Recurrent Neural Networks (RNN) are known to be capable of learning complex temporal sequence. Although RNNs have proven to be successful in many tasks such as text generation and speech recognition, it is difficult for RNNs to learn and train on long temporal sequence. This is due to the vanishing and exploding gradient problem that propagates through the multiple layers of the RNN. This in turn causes the network not to be able to learn effectively. The RNN computes the hidden vector sequence $h = (h_1, h_2, ..., h_T)$ to compute the output vector $y = (y_1, y_2, ..., y_T)$ through iteration of the equations from $t = 1$ to T:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

where, the W denotes weight matrix, b denoting bias vectors and H denoting the recurrent hidden layer function.

LSTM solves above mentioned limitation by containing "memory cells" that allow the network to learn when to forget previous memory states or when to update the hidden states when new information are provided. This allows the network to be better at finding and exploiting long range context. The memory blocks are memory cells that stores the temporal state of the network in addition to special multiplicative units called gates used to control the flow of information. In each memory block, there is an input, an output gate as well as the forget gate. The input gate controls the flow of input activations into the memory cell while the output gate controls the output flow of cell activations into the rest of the network. The forget gate modulates the cell's self-recurrent connections. This was designed to allow the cell to either remember or reset its previous state based on what is needed. Peephole connections are also found in the memory block that connects the internal cells to the gates in the same cell in order to learn precise timing of the outputs.

Using the LSTM architecture, we calculate function H using the following equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$
$$h_t = o_t \tanh(c_t)$$

where σ is the logistic sigmoid function, i_t is the input gate, f_t is the forget gate, o_t is the output gate, c_t is the cell activation vector, h_t being the hidden vector and W being the weight matrices from the cell to get vectors. The additional cells enables the LSTM to learn long-term temporals which traditional RNNs are not capable of learning. In addition, stacking multiple layers of LSTM allows for the modelling of complex temporal sequences. Essentially, the output from the lower level LSTM layer will be the input to upper LSTM layer. The input layer will propagate through a fully connected to the layer above it through a feed forward network. In this work, we utilised three LSTM stacks with 100 hidden units each and input sequences of data for 100 seconds. The loss function we used was a mean-square loss function.

B. Cumulative Sum (CUSUM)

Based on the prediction of the LSTM-RNN, we then calculate the difference between the predicted outputs and the actual sensor data. The idea is to identify any deviations between the actual sensor data and the outputs of the trained model that predicts what the ideal sensor value should be under normal behaviour. Instead of identifying thresholds at each sensor, we apply the CUSUM method to detect the deviations that corresponds to anomalies. CUSUM calculates the cumulative sum of the sequence predictions to detect small deviations over time thus reducing the number of false positives. We calculate both the positive and negative change as the differences between the predicted outputs and the sensor data can be either. We calculate the CUSUM using the following formula:

$$x_0 = 0,$$
$$SH_i = MAX(0, x_{i-1} - Target - b)$$
$$SL_i = MIN(0, x_{i-1} - Target + b)$$

where *SH* calculates the high cumulative sum and *SL* calculates the low cumulative sum. *Target* is pre-defined as the safety limit with *b* being the allowable slack. Hence, if SH_i is greater than $Target + b$ or SL_i is lesser than $Target - b$, then we say that that the data has veered off the defined target. We also include a Upper Control Limit (UCL) and Lower Control Limit (LCL) to act as boundary control to determine that an anomaly has occurred, i.e. if *SH* is greater than UCL or if *SL* is lower than LCL, we determine that an anomaly is detected. In this work, we define *target* as 0.05 and *b* as 0.05 multiplied by the standard deviation of the data. The UCL and the LCL were empirically defined through the validation data from the data set (discussed in Section III and IV).

III. Dataset

In this experiment, we used the SWaT Dataset[1] [6]. SWaT is a fully operational scaled down water treatment plant that is capable of producing 5 gallons/minute of filtered water. SWaT is a six-stage filtration process that mimics a large modern water treatment plant. For more information about the various treatment process, we refer the reader to the SWaT website[2].

The dataset consists of seven days of normal continuous operation and four days with attack. A total of thirty-six attacks were conducted during the four days, and is the most updated and complex open source dataset till date. The data consists of all the sensors and actuator values over the

[1] http://itrust.sutd.edu.sg/dataset/

[2] http://itrust.sutd.edu.sg/research/testbeds/secure-water-treatment-swat/

said duration. However, as a proof of concept, only the data comprising of normal behaviour pertaining to Process 1 (P1) of SWaT was used for training and validating the model in this paper.

In process P1, water can flow into the raw water tank from either the city water supply system, or RO process (P6) after the filtration process. The water from the water supply system is controlled via a motor valve, MV-101. The valve is opened when the water level goes below a predefined low (L) threshold, and it is closed when it reaches a predefined high (H) threshold. The rate of the inflow into the water tank is measured by the flow indicator, FIT-101. P-101 is a pump that is turned on when the water level drops below L in the UF tank (P3), and it is turned off when the water level rises above H in the Ultra Filtration (UF) tank in P3. It can also be turned off when the raw water tank level drops below L, or the flow indicator, FIT-201 (in P2), drops below a certain predefined threshold. Pump P-102 is a redundant pump that goes into operation when pump P-101 fails.

In total, 496,800 samples were used to model the following sensors; FIT101, LIT101, MV101, P101 and P102.

A. Attack Scenarios

The attack dataset was used to identify anomalies in P1. The attacks generated in the dataset were modelled after Adepu et al. ([2], [1]). Their attack model considers the intent space of an attacker for any given CPS. As our model is only trained to model the normal behaviour of P1 in the system, we only take into consideration the attacks pertaining to P1. As listed in Table I, the attack duration depends on the kind of attack. Some attacks are consecutively within a 10 minutes gap of each other, while some of the attacks are performed by leaving time for the system to stabilise.

All the attacks are carried out by fooling the Programmable Logic Controller (PLC) at each process into believing the sensor information it is being sent is genuine, in other words, spoofing the values. Below are some examples of the different type of attacks and how they affect the system.

Single Stage Single Point (SSSP) attacks focus on a single attack point within the same stage (P1 in this case). In attack A1, the intent is to overflow the tank. As described in Table I, the system's start state for A1 shows that MV101 is closed. The attack was carried out by fooling the PLC to open MV101 when it should be in fact closed. This attack causes water to enter the tank when it is already at high level thus overflowing the tank. Fig. 1 shows that attack A1's intent was realised as we see an increase of 100mm in the water storage tank as reported by sensor LIT101.

In another SSSP attack, A3, the attacker's intent was to underflow the tank. This attack fools the PLC to think that the water level (LIT101) is increasing by 1mm every second but is in fact decreasing. This attack causes P101 to continuously pump out water to the next process when water do not exist; potentially damaging the pump. As described in Fig. 1, this attack forces the PLC to react accordingly by turning off MV101.

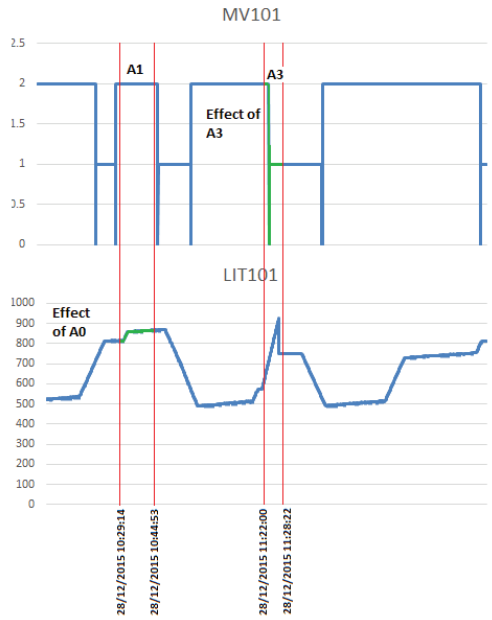

Figure 1: Attack A0: MV101 under attack

Single Stage Multi Point (SSMP) attacks focus on multiple attack points within the same stage (P1). In A7, the attacker's intent here is to stop the outflow of the tank in order for it to overflow. In this scenario, both the pumps P101 and P102 are turned off to stop water from exiting the tank. Over time, this will cause the tank to overflow where LIT101 displayed a sharp increase of 300mm in the dataset over the attack duration of 8 minutes as opposed to 89mm under normal operations.

Multi Stage Single Point (MSSP) attacks focus on single attack points within multiple stages (P1 and P3 in this case). In A9, the attacker's intent is to underflow tank in P1 and overflow tank in P3. This was performed by fooling the PLC to turn on P101 in order to continuously provide water to the tank in P3. At the same time, LIT301 was set to be at 801mm. The actions of this attack resulted in the tank in P1 under-flowing and the tank in P3 overflowing.

Multi Stage Multi Point (MSMP) attacks focus on multiple attack points within multiple stages (P1 and P3). Although the attacker's intention is similar to A9, this attack shows that it is possible to attack a CPS through multiple attack points across different stages. In A10, in order to underflow the tank in P1 and overflow the tank in P3, PLC is fooled by turning on P101 and believing that LIT101 is at a level of 701mm. However, in reality, LIT101's actual value is significantly lower than 701mm. This causes the pump to be on even though all the water has already been delivered to P3. In addition to attacking points in P1, MV201 in P2 was also turned on. This effectively means that P3 is receiving water from both P1 and P2 causing a overflow of P3 over time.

Type of Attack	Attack ID	Attacker's Intent	Start State of System	Description of Attack
Single Stage Single Point Attacks (SSSP)	A1	Overflow tank	MV-101 is closed	Open MV-101
	A2	Burst the pipe that sends water between process P1 and process P2	P-101 is on where as P-102 is off	Turn on P-102 in addition to P-101 to flood the pipe
	A3	Underflow the tank and damage P-101	Water level is between L and H	Increase water level by 1 mm every second
	A4	Stop outflow of tank	P-101 is on	Turn P-101 off
	A5	Tank underflow and damage P-101	Water level is between L and H	Set LIT101 to above H threshold
	A6	Overflow Tank	Water level is between L and H	Set LIT101 to less than LL
Single Stage Multi Point Attacks (SSMP)	A7	Tank overflow	MV-101 is open; LIT-101 between L and H	Keep MV-101 on continuously; Value of LIT-101 set as 700 mm
	A8	Stop outflow of tank	P101 is on; P102 is off	Turn P101 off; Keep P-102 off
Multi Stage Single Point Attacks (MSSP)	A9	Underflow tank in P1; Overflow tank in P3	P-101 is off; P102 is on; LIT-301 is between L and H	P-101 is turned on continuously; Set value of LIT-301 as 801 mm
Multi Stage Multi Point Attacks (MSMP)	A10	Underflow tank in P1; Overflow tank in P3	P-101 is off; MV-101 is off; MV-201 is off; LIT-101 is between L and H;,LIT-301 is between L and H	Turn P-101 on continuously; Turn MV-201 on continuously; Set value of LIT-101 as 700 mm;

Table I: Attack descriptions in P1

Sensor	Upper Control Limit	Lower Control Limit
FIT101	4	-10
LIT101	10	-10
MV101	8	-11
P101	8	-11
P102	1	-7

Table II: Upper and Lower Limit Identification

IV. Experiment and Analysis

A. Pre-processing

In this experiment, we use all the values of the sensors and actuators in P1 as features for training the model. In essence, the value from LIT101, FIT101, MV101, P101 and P102 were used. All the sensors and actuators in P1 are treated as a numeric attribute. We normalise each feature (all sensors & actuators) by removing the mean and scaling to unit variance. In this work, both the mean and scaling are applied on each individual feature as they have different range. For example, FIT101 typically ranges from 0 to 2.6 whereas LIT101 ranges from 0 - 900. Hence, all the data was normalised to be on the same scale, to prevent the domination of any features over the others.

Subsequently, the normal behaviour data was divided into 80% for training and 20% for validation.

B. Training

Training of the LSTM was performed on a XEON class server with 64GB of RAM using a Nvidia 2GB 750ti GPU. Training was made to run until the validation loss stopped decreasing or until it hit its maximum iteration of 200. Using the entire training set (normal behaviour data) for P1, the model took approximately 24 hours to train. After the model is trained, we used the validation data to obtain the UCL and the LCL for the CUSUM as illustrated in Table II.

C. Results

In this section, we discuss the anomalies that were detected by the proposed method. In total, 9 out of the 10 attacks were detected. We describe the detection of attacks A1 to A3 through Fig. 2. In attack A1, the attack occurred at the motorised valve, MV101. During this attack, the operation of the testbed became abnormal and caused an increase in FIT101 and LIT101. Due to the level of the water tank increasing, this lead on to P101 being turned on by the PLC. All these anomalies are detected by the prediction model and the CUSUM as illustrated in Fig. 2

In the next attack, A2, P102 was turned on in addition to P101 with the intention to burst the pipe. This lead to the sudden decrease in water (LIT101) in the water tank. Due to the decrease of water in the water tank, the PLC reacted erroneously by turning on MV101 which caused FIT101 to be activated as well. Similarly, this was captured in Fig. 2.

In A3, LIT101 was attacked to fool the PLC into believing that the water level is increasing. This anomaly was detected at LIT101, P101, MV101 and FIT101. This is because LIT101 was attacked to reflect a level of 700mm, this caused P101 to be activated to pump water to the next process. While the water is outgoing, MV101 was turned on to release water into the P1 tank.

Attacks A5 and A6 were similar in nature, LIT101 were attacked to either set LIT101 to above the H or below the L threshold. As illustrated in Fig. 3, these anomalies were detected in FIT101, LIT101, MV101 as well as P101.

Of all the six SSSP attacks, only attack A4 failed to be detected. This is because the attack was to turn off P101 with the intention to stop the outflow of the tank in order to overflow it. However, in this case, the PLC reacted by turning on P102 which is considered normal behaviour. As observed in Fig. 3, there are three false positives (FP). Despite looking through all the attack listings for the entire SWaT dataset, we were unable to identify any attacks within the other processes during the stipulated time of F1. However, F1 consistently existed in MV101, FIT101 as well as LIT101 with rather high CUSUM values. We believe that the anomaly is genuine and was the result of a previous attack. Similarly for F2 and F3, these consistently appeared in MV101, FIT101, LIT101 and P101. Again, we were not able to explain why an anomaly was detected and deduced that it most likely appeared due to

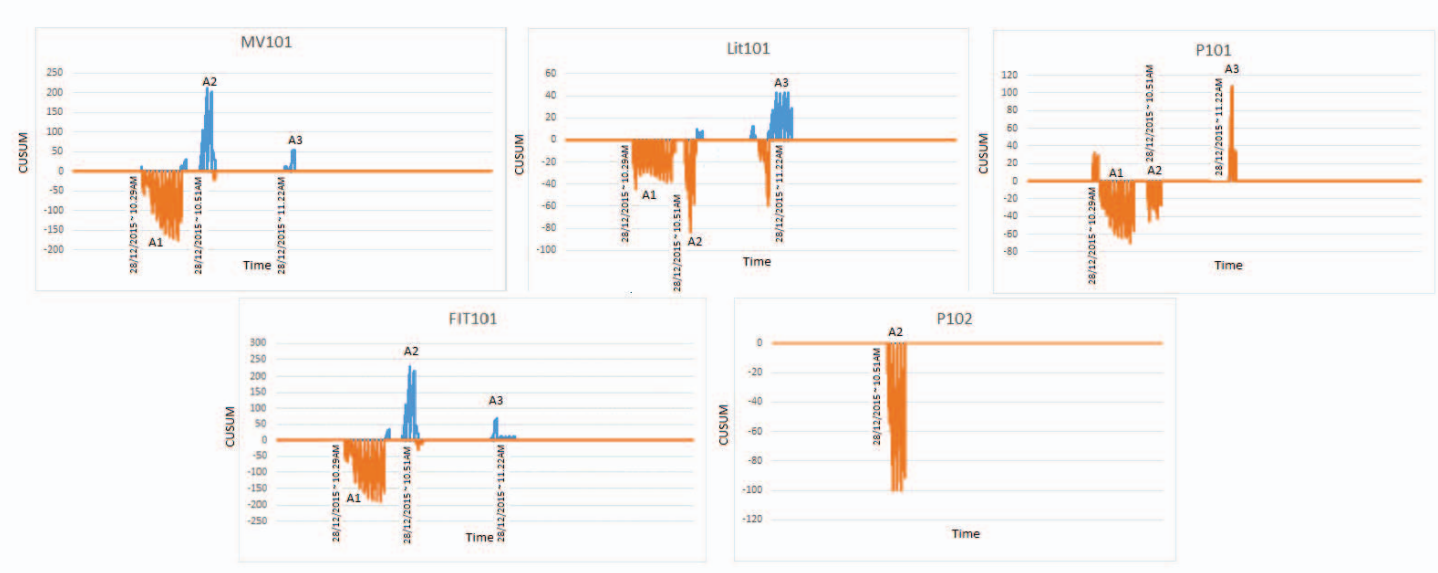

Figure 2: Detected Attack I

a previous attack.

The proposed method was also successful in detecting SSMP attacks. As illustrated in Fig. 3, attack A8 was detected in P101 and P102 where these two pumps were deliberately both turned off. In Fig. 4, the anomaly caused by attack A9 was detected in LIT101 and P101. Also in Fig. 4, we identified one additional anomalies and one FP (FP4). We cross referenced the time of the detected anomaly, AP1 and found that it corresponded with attacks that were targeted at P2 (AP1). This leads us to believe that attacks targeted for different process can ultimately affect other processes within the CPS. FP1 exhibited high CUSUMs in MV101, FIT101, P101 as well as LIT102. Similarly to F2 and F3, we believe that the anomaly was caused by two earlier attacks that occurred within 20 minutes of each other on P2 and P4.

MMSP and MSMP attacks were also successfully detected. As illustrated in Fig. 4, anomalies caused by the attack A9 were detected across all the sensors in P1. The nature of the attack was to continuously turn on P101 and set LIT301 (P3) to 801mm. An anomaly was detected in P102 because it was forced to turn off as P101 was turned on. As P101 was continuously turned on with the intention of underflowing the tank in P1, LIT101 was detecting the anomaly in which the water level was decreasing. As there was no water in P1, MV101 was turned on by the PLC to provide water to the tank, thus activating FIT101 as well. All these were not consistent with normal behaviour and were detected successfully. A10's attack intention is similar to that of A9 where the Anomalies were also detected throughout all the five sensors in P1.

D. Limitations

The current proposed approach is only applicable to P1. This is because we were unable to train all the sensor data due to the vast amount of data involved and our limited infrastructure. Due to this reason, we are also unable to validate the FPs as the model is only restricted to P1. As future work, we would be proposing a distributed learning approach to correlate and learn the entire dataset. In addition, we would test our approach to the entire system to properly correlate and analyse the FPs.

V. Conclusions

The securing of CPS from cyber-attacks is a high priority for many governments. While many IDS exists, they focus mainly on network traffic. In addition, majority of behaviour based approaches are on specification or signature-based techniques. This paper proposes an unsupervised learning approach for anomaly detection in the area of CPS. Furthermore, through the experiments in this paper, we derive that attacks occurring in other processes can be detected among each other. We successfully demonstrated that this method is effective through the use of the SWaT dataset obtained from a complex testbed.

Acknowledgment

This work was supported by the National Research Foundation (NRF), Prime Ministers Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-40).

References

[1] S. Adepu and A. Mathur. Generalized attacker and attack models for cyber-physical systems. In *The 40th IEEE Computer Society International Conference on Computers, Software and Applications*, 2016.

[2] S. Adepu and A. Mathur. An investigation into the response of a water treatment systemto cyber attacks. In *In Proceedings of the 17th IEEE High Assurance Systems Engineering Symposium (in Press)*, 2016.

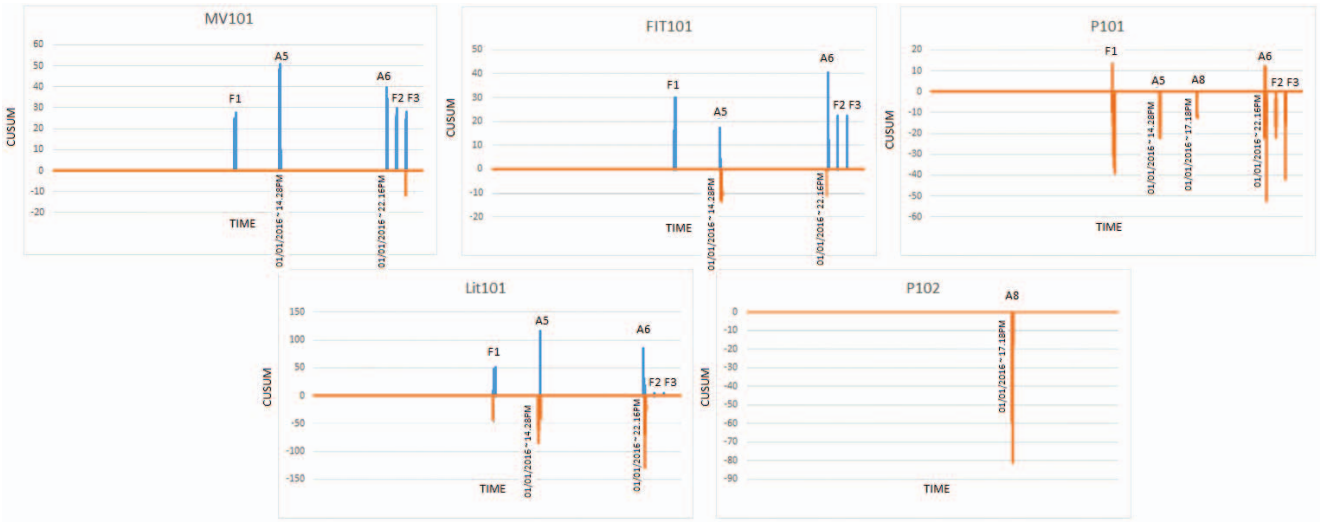

Figure 3: Detected Attack II

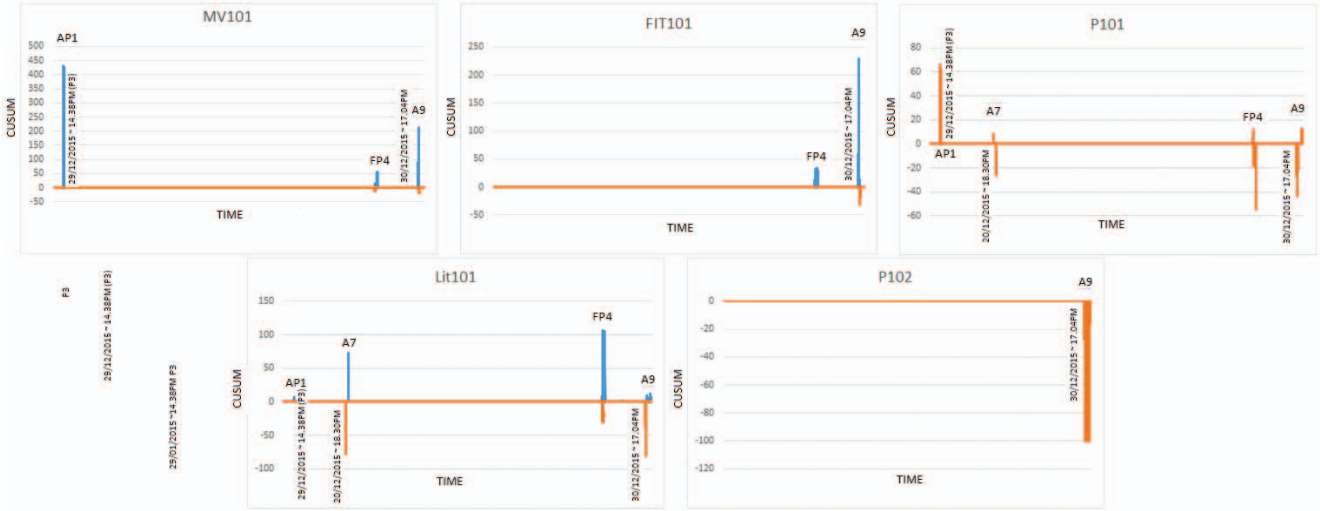

Figure 4: Detected Attack III

[3] O. Al-Jarrah and A. Arafat. Network intrusion detection system using neural network classification of attack behavior. *Journal of Advances in Information Technology Vol*, 6(1), 2015.

[4] J. M. Beaver, R. C. Borges-Hink, and M. A. Buckner. An evaluation of machine learning methods to detect malicious scada communications. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, pages 54–59, Dec 2013.

[5] N. Falliere, L. O. Murchu, and E. Chien. W32.stuxnet dossier.

[6] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur. a dataset to support research in the design of secure water treatment plants khurram critis. *To appear in The 11th International Conference on Critical Information Infrastructures Security*, Oct 2016.

[7] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan. Machine learning for power system disturbance and cyber-attack discrimination. In *Resilient Control Systems (ISRCS), 2014 7th International Symposium on*, pages 1–8, Aug 2014.

[8] K. N. Junejo and D. Yau. Data driven physical modelling for intrusion detection in cyber physical systems. In *CVolume 14: Proceedings of the Singapore Cyber-Security Conference (SG-CRC)*, pages 43 – 57, 2016.

[9] R. M. Lee, M. J. Assante, and T. Conway. Analysis of the cyber attack on the ukrainian power grid, 2016.

[10] X. Li and X. Wu. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4520–4524. IEEE, 2015.

[11] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal. Long short term memory networks for anomaly detection in time series. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 89–94, 2015.

[12] P. Nader, P. Honeine, and P. Beauseroy. lp -norms in one-class classification for intrusion detection in scada systems. *IEEE Transactions on Industrial Informatics*, 10(4):2308–2317, Nov 2014.

[13] P. Nader, P. Honeine, and P. Beauseroy. Mahalanobis-based one-class classification. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sept 2014.

[14] J. Slay and M. Miller. *Lessons learned from the maroochy water breach*. Springer, 2008.