Anomaly Detection in Cyber-Physical Systems in Fog Environment

Anvika D Shriyan[1], Khyati K Kaneria[2], Navya M R[3], Shrut Kriti[4], Prof. Arun Vikas Singh[5]
[1,2,3,4,5]*Dept. of Computer Science and Engineering, PES University, Bengaluru, India*

Abstract **- A cyber physical system (CPS) is an intelligent system which uses computer algorithms to monitor and control the functions of the system. Smart healthcare, smart electric grids, and aircraft autopilot systems are some examples of CPSs. Cyber Physical Systems have become highly integrated in the modern world. The importance of safeguarding these systems grows as this connection advances. Attacks against Cyber Physical Systems components can result in faulty sensing and actuation, catastrophic damage to physical items, and safety concerns. Anomaly detection can be used to detect and stop such attacks by monitoring system activity and classifying it as normal or anomalous. Machine learning algorithms have been proposed to build anomaly detection systems to hinder attacks on CPSs. However, the intricate interdependencies among numerous variables and a lack of labelled data in these systems render typical supervised machine learning methods ineffective. In this report, we propose an unsupervised anomaly detection method based on Generative Adversarial Networks (GANs), using Long-Short-Term-Memory Recurrent Neural Networks (LSTM-RNN) as the base models (namely, the generator and discriminator) in the GAN framework to capture the temporal dependencies of variables. Because intrusions must be stopped before the server is infected, anomaly detection in Cyber Physical Systems has strict latency constraints. Current anomaly detection models, despite having good detection rates, are excessively slow and unsuitable for latency-constrained situations. Hence, we plan to propose a model with lower latency, by using fog computing. Fog computing helps to bring computation power closer to end nodes, which helps to meet anomaly detection's low latency standards.**

INTRODUCTION

Cyber Physical Systems are a combination of computation, networking, and physical processes. CPS contains feedback loops of embedded computers monitoring and controlling the physical processes and physical processes affecting the computations. Figure shows the various parts of a CPS. Sensors, actuators, programmable control logic units, and communication devices are generally included in these systems. The qualities of such a system are decentralized management and control, efficiency, high availability, scalability and autonomy. Healthcare devices and systems, traffic control and safety, automotive systems, control systems, energy efficiency, environmental monitoring, instrumentation, infrastructure control, autopilot and avionics, power generation, water management, and communications systems all present CPS with unprecedented opportunities. Because the bulk of cyber physical systems have applications that have safety concerns, any disruption caused by failures or intentional cyber-attacks will result in catastrophic destruction to the physical infrastructure under control and the people impacted by them. As a result, preventing anomalies and attacks from compromising CPS is crucial.

Cyber physical systems are becoming an integral part of distributed systems, control systems, sensor automotive systems. As cyber physical systems are becoming an integral part of systems and hence it is also prone to attacks. So it is necessary to detect any possible intrusion or attack on the system before it compromises the system.

We plan to build a GAN model which can detect an attempt of intrusion in various cyber physical systems. There are instances of abnormal behaviour when a malicious attack starts on the system. The plan is to detect initial signs of any intrusion in the system and detect the attack in an accurate and faster manner by the use of fog architecture.

MATERIALS AND METHOD

Data

The SWaT dataset was chosen to majorly train and test our model. Other than SWaT various other datasets like SKAB, MSL, DAMADICS and SMAP were used to compare our GAN model with other existing

models.The dataset NSL-KDD was initially used when we were building the decision tree model.

SWAT Dataset

The data collection process lasted for a total of 11 days. SWaT was functioning non-stop 24 hours/day, during the entire 11-day period. SWaT was run without any attacks during the first seven of the 11-days. Attacks were launched during the remaining four days. The dataset describes the physical properties of the testbed in operational mode. In total, 946,722 samples comprising 51 attributes were collected over 11 days. Data capturing the physical properties can be used for profiling cyberattacks. Table 5 describes the different sensors and actuators in SWaT that served as source of the dataVarious attack scenarios were implemented on the testbed. These attacks were of various intents and lasted between a few minutes to an hour. Depending on the attack scenario, the system was either allowed to reach its normal operating state before another attack was launched or the attacks were launched consecutively.

DAMADICS

The DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) benchmark consists of real process data from the Lublin Sugar Factory as well as a simulator to generate artificial faults. Here we use only the real dataset with induced faults in the industrial system, available publicly. The dataset consists of data for 25 days of operation, from Oct-29 to Nov-22, 2001 of 3 benchmark actuators - one each located upstream and downstream of evaporator station, and the third controlling flow of water to the steam boiler system. Artificial faults were induced on Oct-30, Nov-9, Nov-17 and Nov-20. Unlike SWaT the train and test splits for normal and test operation were not provided. We chose train-test splits such that the test is entirely after the train as would be expected in a real scenario, the train is continuous, and the train contains no anomalies. Accordingly, we used the data from Nov-3 to Nov-8 (6 days) as the training set and data from Nov-9, Nov-17 and Nov-20 (3 days) as the test set. From this, the first 10800 points from the training set were dropped as the system appeared much more unstable than the rest of the training set, potentially from the anomaly induced earlier on Oct-30. In addition, the initial part of the test set appears quite unstable across multiple channels even though no anomaly is recorded. Therefore we also drop the first 45000 points from the test set.

MSL and SMAP

These are expert-labeled datasets from real anomalies encountered during the operation of two spacecraft - Soil Moisture Active Passive (SMAP) satellite and the Mars Science Laboratory (MSL) rover, Curiosity. Unlike the other datasets, each entity in MSL and SMAP consists of only 1 sensor, while all the other channels are one-hot-encoded commands given to that entity. We use all channels as input to the model, but the model error of only the sensor channel is used for anomaly detection. The total number of variables is 1375 and 1485 for SMAP and MSL respectively, making these much larger than the single-entity datasets in terms of number of variables. The data is however divided into 55 and 27 entities respectively. The authors provide train-test splits so that for the first anomaly encountered in the test at time t, the training set is from time t-5 days to t-3 days (if available), and the test set goes from t-3 days to t+2 days. The data is sampled each minute, so the training set is much shorter than other datasets.

SKAB

The Skoltech Anomaly Benchmark testbed consists of a water circulation system and its control system, along with a data-processing and storage system. Examples of the type of anomalies induced include partial valve closures, connecting shaft imbalance, reduced motor power, cavitation and flow disturbances. Train and test splits are provided by the authors.

METHODOLOGY

GAN Architecture

In order to handle the time series data we implemented LSTM-RNNs in the generator and the discriminator. LSTM's help in resolving the vanishing gradient problem of typical RNN'S and also helps in remembering past data in memory with the help of back propagation. The generator model takes in account the SWaT data and it tries to replicate the distribution whereas the discriminator with proper training distinguishes the real data from the forged data. Two types of losses are calculated which are subsequently used to calculate Discrimination and

Reconstruction Anomaly Score (DR-Score).The reconstruction loss is the difference between the testing samples created by the generator and the actual samples present and the discrimination loss is a parameter that checks and maximises a parameter which takes in account and maximises a function whenever the real data is classified as fake data and vice versa.

Anomaly Detection using GAN Architecture

We divide the dataset into training and testing dataset into streams where there are T streams and M measurements for each stream, the testing dataset X(test) comprises T streams and N measurements and we assign binary values 0 for normal behaviour and 1 for anomalous behaviour. We apply a sliding window and create small window sizes and divide the multivariate time series. We use the standard GAN loss function min-max GAN loss for calculating the generator and the discriminator loss.

We maximize the below function in order to classify as real and fake data:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right]$$

log(D(x)) this parameter is used to specify whether the generator is rightly classifying the real image. log(1-D((G(z))) maximising this function label the generator generated images and label the fake images. The data windows are divided into small sub sequences (Z) and the training dataset is (X). So now we feed X and Z to the GAN model in order to train the generator and discriminator and we maximise and minimise the function.

We run training iterations and combine the trained discriminator Drnn and the trained generator Grnn and a combined anomaly score is calculated which comprises of discrimination and reconstruction score (DR Score). The DR score in case anomaly is detected is non-zero while when the anomaly is detected is zero.

DR-Score

DR Score is calculated by taking in consideration both the generator and the discriminator. We can detect the anomaly in two ways by taking in consideration the generator and the discriminator.

Latency reduction using Fog Architecture

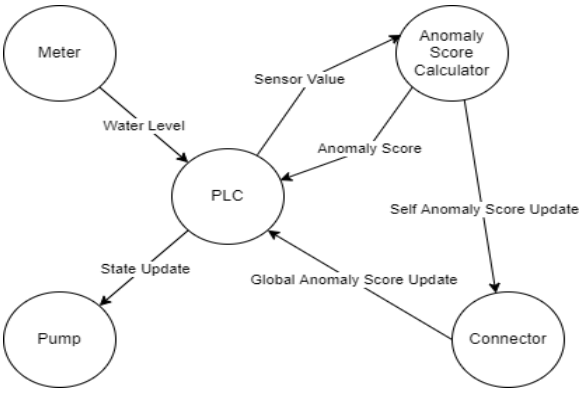

Fig 1: Application Model representing the module interaction

We used the iFogSim Simulator in Eclipse IDE for Fog Simulation. It is an extension of CloudSim and an open-source Java based toolkit which computes latency, energy consumption and the operational costs for a fog network. It also facilitates importing topologies and helps in the comparison of resource management techniques and calculates the cost for each configuration.

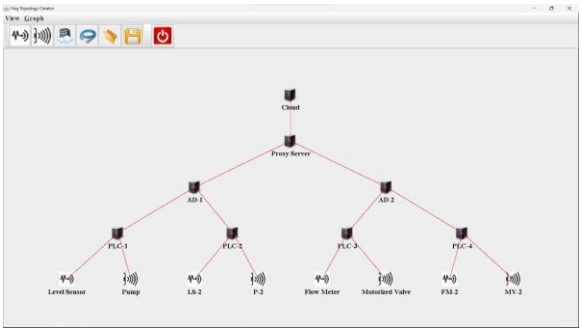

Fig 2: The proposed fog-based topology for Anomaly Detection

The Fog topology has five levels. The topmost level is the cloud layer. It is connected to the proxy server which helps in load-balancing. The proxy-server is further connected to the Anomaly Detectors. Anomaly detectors are extended to the programmable logic controllers. The number of anomaly detectors and the programmable logic controllers in the topology can be configured based on the requirement by modifying the code. The variables numOfADs and numOfPLCsPerDet need to be modified respectively. The PLCs are extended to the sensors and the actuators. Here, the sensors used are level sensors and flow meters and the actuators are Pump and Motorized valve. The level sensor or the flow meter senses the

input and sends it to the PLC which then sends the signal to the Anomaly detector where the anomaly score is calculated using the anomaly score calculator module and the PLC is updated while the data is updated in the Cloud. The cloud layer stores all the logs and does not involve itself in computing the anomaly score. The PLC sends the signal to the actuators namely pump or the motorized valve and the required action is performed.

If a cloud based architecture is used for performing Anomaly Detection, the fog devices are not included instead the sensors and actuators are directly connected to the proxy server.

RESULTS AND DISCUSSION

SWaT Dataset

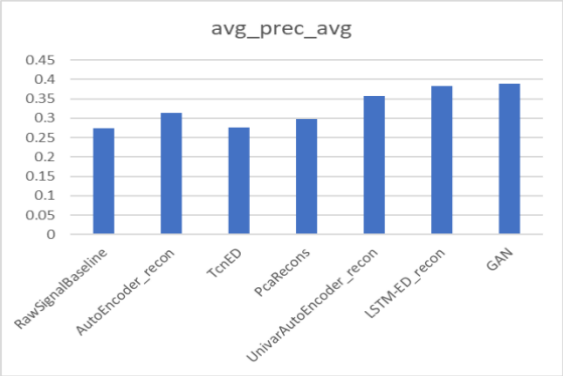

Fig 3: Average precision comparison

The figure compares the precision of 7 models by taking the average over the entire test which was performed using the previously mentioned SWaT dataset. Higher precision means that an algorithm returns more relevant results than irrelevant ones. As observed from the figure our GAN model has a slightly better precision when compared to other models.

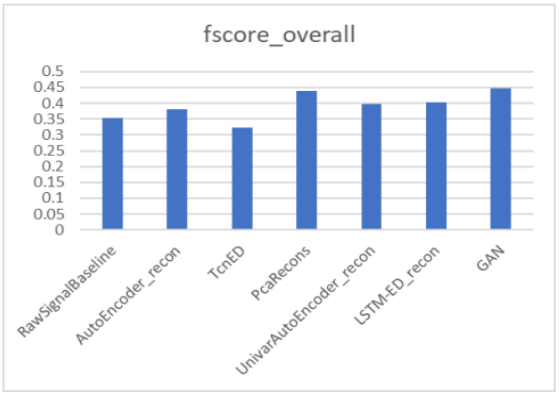

Fig 4: F1 score comparison

The figure compares the F1 score of 7 models tested using the previously mentioned SWaT dataset. F1 score is the harmonic mean of precision and recall. It is commonly used as a measure of accuracy when testing the performance of anomaly detection models.

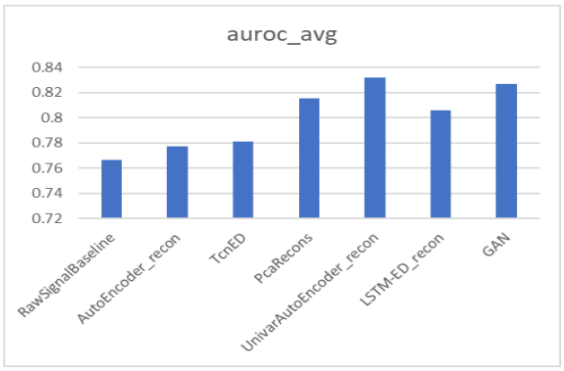

Fig 5: Area under the ROC curve comparison

The Figure compares the Area under the ROC curve of 7 models tested on the SWaT dataset. As we can observe in the figure, the raw baseline model performs the worst because it does not perform any computation. The Area Under the Curve (AUC) is the measure of the ability of a model to differentiate between classes and is used as a summary of the ROC curve.

Fog Architecture Results

Table 1: Latency results obtained after simulating various configurations in iFogSim

Configurations (AD*PLC)	Latency in Fog (in ms)	Latency in Cloud (in ms)
1*1	17.897	220.045
1*2	19.072	220.278
2*2	19.067	220.525
4*2	19.069	221.585
4*4	33.14	224.525
8*4	33.433	2995.061

The values obtained shown in the table above are plotted and the results are represented in a line graph shown in Fig. The configurations represent the number of Anomaly Detector and Programmable Logic Controller devices in the topology which can be modified based on the requirement. The number of

sensors can be deduced by multiplying both the values i.e.,
Number of sensors = Number of Anomaly Detectors * Number of Programmable Logic Controllers.

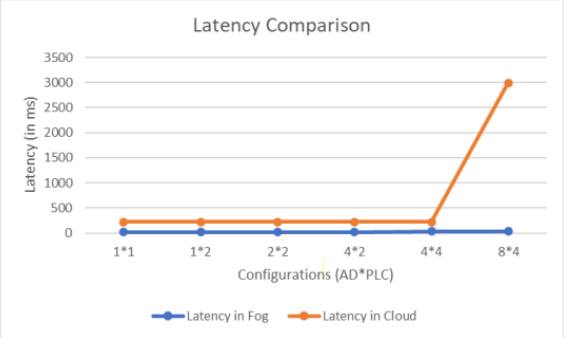

Fig 6: Comparison of latencies for various configurations

It can be noted that there is a surge in the latency of cloud when the configuration reaches 8*4. This is due to the network congestion issues.The load on the cloud server increases as all the operations are performed solely in the cloud whereas in the Fog architecture, the operations are divided almost equally among the devices leading to a line graph which is nearly constant.

Table 2: Network Usage results obtained after simulating various configurations in iFogSim

Configurations (AD*PLC)	Network usage in Fog (in kB)	Network usage in Cloud (in kB)
1*1	2441	20055
1*2	4205.1	40422.2
2*2	7414.9	83550
4*2	13840.3	178306
4*4	32262.8	379259.2
8*4	63616.6	741073.2

The data in Table represents the network usage for various configurations of cloud and fog topologies. It can be seen from the Fig. that the network usage in cloud is significantly higher than that of fog. This is because, when the number of sensors increase, the traffic towards the cloud server increases thus resulting in higher network usage. In case of the fog based configurations, each sensor is assigned to a PLC, hence the network usage decreases, resulting in a greater and consistent throughput for the other requests when compared to cloud based scenarios.

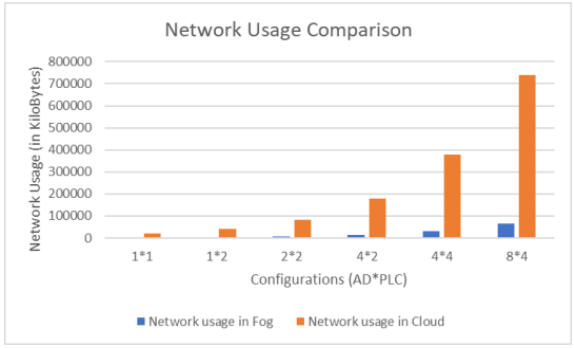

Fig 7: Comparison of network usage for various configurations

CONCLUSION

The fog architecture reduces the delays by provisioning a sensor to a fog device and reducing the load on a central server and helps in regulating a reliable real time anomaly detection architecture. Fog computing is essential in scenarios where a quick response is necessary. Fog architecture significantly reduces the latency and the network usage when compared to the cloud architecture and hence is found to be reliable for real-time data processing applications and other scenarios where latency is a concern.

REFERENCE

[1] E. A. Lee, “CPS foundations,” in Proc. 47th ACM/IEEE Des. Autom. Conf., Jun. 2010, pp. 737–742.
[2] A. Cardenas, S. Amin, and S. Sastry, “Secure control: Towards survivable cyber-physical systems,” in Proc. 28th Int. Conf. Distrib. Comput. Syst. Workshops, Jun. 2008, pp. 495–500.
[3] Y. Tan, S. Goddard, and L. C. Perez, “A prototype architecture for cyberphysical systems,” ACM SIGBED Rev., vol. 5, no. 1, Jan. 2008, pp. 1–2.
[4] I. Lee and O. Sokolsky, “Medical cyber physical systems,” in Proc. 47th ACM/IEEE Des. Autom. Conf., Jul. 2010, pp. 743–748.
[5] M. D. Ilic L. Xie, U. A. Khan, and J. M. F. Moura, “Modeling of future cyber-physical energy systems for distributed sensing and control,” IEEE

Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 40, no. 4, pp. 825–838, Jul. 2010.
[6] R. (R.) Rajkumar, I. Lee, L. Sha, and J. Stankovic, “Cyber-physical systems: The next computing revolution,” in Proc. 47th ACM/IEEE Des. Autom. Conf., Jul. 2010, pp. 731–736.
[7] S. Han, M. Xie, H. -H. Chen and Y. Ling, "Intrusion Detection in Cyber-Physical Systems: Techniques and Challenges," in IEEE Systems Journal, vol. 8, no. 4, pp. 1052-1062, Dec. 2014, doi: 10.1109/JSYST.2013.2257594.
[8] P. Freitas de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. Gondim Santos, D. Macêdo and C. Zanchettin, "Intrusion Detection for Cyber–Physical Systems Using Generative Adversarial Networks in Fog Environment," in IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6247-6256, 15 April15, 2021, doi: 10.1109/JIOT.2020.3024800.
[9] Hassan, Syed & Ahmad, Ishtiaq & Ahmad, Shafiq & Alfaify, Abdullah & Shafiq, Muhammad. (2021). Remote Pain Monitoring Using Fog Computing for e-Healthcare: An Efficient Architecture. Sensors. 20