An Investigative Study on Minimising Security Attacks using Deep learning in a Cyber Physical System

Manas Kumar Yogi*[1], Dr. A.S.N. Chakravarthy[2],
Asst.Prof. CSE Dept.[1].,
Professor, CSE Dept.[2], Pragati Engineering College (Autonomous),
Surampalem,A.P.[1],
JNTUK-University College of Engineering Vizianagaram,A.P.[2]

***Abstract*-Irregularity identification is essential to guarantee the security of cyber-physical systems (CPS). Be that as it may, because of the expanding intricacy of CPSs and more modern assaults, traditional inconsistency recognition strategies can't be straightforwardly applied to defeat such issues, which additionally need area explicit information and handle the developing volume of information. Profound learning-based peculiarity discovery techniques have been proposed to accomplish solo recognition in the period of CPS huge information. In this paper, we audit best aspects of these deep learning strategies in CPSs. We propose a scientific categorization as far as the sort of peculiarities, techniques, and usage and assessment measurements to comprehend the fundamental properties of current strategies. Further, we use this scientific categorization to recognize and feature new qualities and plans in every CPS area. We sum up top notch of openly accessible datasets for preparing and assessment. We additionally talk about our discoveries, the restrictions of existing investigations, and potential headings to improve security in CPS using deep learning techniques.**

Keywords-Deep Learning; Anomaly detection; Cyber Physical Systems; Security; Privacy

I. INTRODUCTION

Cyber-physical systems (CPS) are increasingly being deployed in basic infrastructures. The CPS market is expected to expand by 9.7% each year, which will reach $9563 million by 2025. Prominent uses of CPS include mechanical control systems (ICS), shrewd grid, intelligent transportation systems (ITS), and aerial systems. CPSs have evolved to be complex, heterogeneous, and integrated to provide rich functionalities. However, such characteristics additionally expose CPSs to broader threats. According to FireEye's report, insiders, ransomware, market control, etc. are among the top assault types in ICS. Recent incidents (e.g., Stuxnet, Ukraine power grid outage, auto-driving crashes, robot malfunction) have indicated that sophisticated and stealthy assaults (and blames) can result in calamitous consequences to the economy, environment, and even living souls. Subsequently, it is central critical to ensure the security of CPSs. To detect assaults and unexpected errors in CPSs, anomaly detection methods are proposed to mitigate these threats. For example, rule, state estimation (e.g., Kalman filter), statistical model (e.g., Gaussian model, histogram-based model) based methods are utilized to learn normal status of CPSs [64]. However, these methods usually require expert knowledge (e.g., operators manually extract certain rules), or need to know the underlying dispersion of normal information. Machine learning approaches don't rely on space specific knowledge. Yet, they usually require a large amount of labelled information (e.g., order based methods). Likewise, they can't capture the unique attributes of CPSs (e.g., spatial-temporal correlation) . Interruption detection methods are dedicated to ensuring network correspondence security. Physical properties (e.g., the noise of engines) are captured to depict the immutable nature of CPSs. Program execution semantics are characterized to protect control systems. However, as CPSs become more complicated what's more, assaults are more stealthy (e.g., APT assaults), these methods are difficult to ensure the overall status of CPSs (e.g., protect multivariate physical measurement) and need more space knowledge (e.g., more components and correlation). Anomaly detection systems need to adjust to capture new characteristics of CPSs. Specifically, we need to answer three research questions:

(1)What are the characteristics of existing approaches? Specifically, the threat model, detection strategies (i.e., input information, neural network design, and anomaly scores), implementation and evaluation metrics of Deep Learning methods are definitely not categorized.

(2) What are the takeaways and impediments of existing work? Are there freely available datasets?

(3) How would we be able to improve Deep Learning methods?

Answering these questions helps to understand the fundamentals of Deep Learning methods, evaluate proposed DLAD models, and explore new arrangements.

II. BACKGROUND

A. Complexity Management

Anomaly detection has developed for various applications, e.g., intrusion detection, fraud detection. In this work, we centre on new research efforts that detect anomalies in CPS with the help of emerging deep learning methods. We can concisely characterize the generic work process of Deep Learning methods. Normally, Deep Learning methods comprise of training and testing phases. At the training phase, a large amount of info information is first collected. Sensor and actuator information, level 0 and level 1 correspondence traffic, and control system logs are regularly used information sources. Different customized information processing approaches are applied to the information, which is then fed to neural network models. Then, the principle commitment of new methods lies in different DLAD models (e.g., RNN, auto encoders, CNN, and customized models) in different application scenarios. Further, DLAD models utilize misfortune capacities to compute differences between yield information from the yield layer and ground truth information. We denote these differences as anomaly scores. There are three types of anomaly scores: (1) Prediction error (2) Reconstruction error, and (3) Predicted labels (details in Section 3.2). Anomaly scores are used to optimize and update DLAD models. At the testing phase, collected or real-time input information is fed to trained models and determine whether the information is an anomaly. As an early effort to review anomaly detection methods, they didn't consider deep learning based methods and didn't include CPS. Item IoT systems have transformed the way people live. For example, emerging keen home applications permit users to interact with home appliances automatically. Program investigation methods are proposed to protect the security and discover vulnerabilities in these applications. Meanwhile, researchers have reviewed anomaly detection methods that utilize the physical properties of CPSs (e.g., the noise of physical devices) [1,2]. Studies in terms of network security of SCADA systems are summarized with an attention on danger assessment techniques [3]. Yet, the techniques didn't include deep learning methods and are conventional, e.g., state estimation, intrusion detection based methods. There is work that studied deep learning-based anomaly detection methods however didn't zero in on CPS [4]. While many of the researchers have investigated utilizations of deep learning methods in CPS, it didn't cover anomaly detection [5]. Some of the researchers have also examined the scientific classification of threats in shrewd home IoT, which did not consider anomaly detection methods [6,7]. At last, few of the researchers have studied information examination approaches that use deep learning methods in IoT [8]. To the best of our knowledge, our work is the principal work that studies deep learning-based anomaly detection methods in CPS, which differs from the above existing surveys.

III. APPLICATION OF DEEP LEARNING FOR INCONSISTENCY EXPOSURE

Since CPSs usually manage basic infrastructure (e.g., ICS, medical devices, and power grid), they are consistently under the threat of different assaults. An attacker who has the motive (e.g., monetary interest, protection theft, and state operations) can lead assaults. These assaults can target different pieces of CPSs:

(1) Network correspondence layer. Field devices (e.g., sensors and actuators) rely on correspondence networks to cooperate with each other. Additionally, sensor values, device status are reported to server farms and control commands are sent by control systems through the network. In this case, level 0 correspondence (C0) and level 1 correspondence (C1) can both be targeted. Note that S2, A2, D1 (contained in C0 and C1 traffic) can likewise be manipulated under these assaults.

We identify three types of assaults:

• Denial-of-service (DoS) assaults: DoS assaults bring a significant threat to the functionalities of real-time applications in CPSs. For example, it would cause a crash of airplane or low traffic use if the ADS-B system is unavailable. Meanwhile, the transmission feature in some CPS correspondence protocols (e.g., the CAN protocol in shrewd vehicle systems) makes the network prone to DoS assaults.

• Man-in-the-middle (MITM) assaults: PSs receive numerous newly designed protocols, which may do not have a well-designed authentication mechanism. Additionally, Ethernet used in CPS can be exploited to direct MITM assaults. Packet content might be manipulated and sensitive information can be leaked through MITM assaults.

• Packet injection. In the event that attackers gain access to the network, they are able to inject a subjective packet to send control command into the system. False control commands can cause severe damage to running devices and even place human lives under danger. For example, a false engine and brake control command might induce an auto accident.

(2) Control system. As the core of one CPS, control systems take sensor values as info and give control signals to actuators or field devices. Due to brutal working environments or limited hardware resources, the protection mechanism may not well-established in charge systems.Once control systems are compromised, information sent to SCADA systems (D1) and commands sent to actuators (A2) can be altered. We discover two types of assaults that target control systems:

• Malware: For the long-term monitoring and information leakage, attackers would place malware in the control system. Moreover, malware can be used to dispatch a stealthy assault (e.g., APT assault) at a certain crucial point in time. Sensor readings can be manipulated by malware. Under certain circumstances, malware may likewise cause physical damage to devices .

• False control signals: Devices operate deviating from regular working status when receiving false control signals. Wrong operations shorten the working life of devices and can even damage devices directly. Attackers usually conceal their unauthorized access to the system and send false control commands at a crucial time point.

Shortcomings: The complexity of systems and heterogeneity of devices lead CPSs to generate unexpected flaws. For example, modern control systems regularly comprise of multiple stages and a ton of components in each stage. Numerous devices operate in a cruel environment (e.g., high dampness or temperature). Additionally, mechanical parts are vulnerable to scraped spot and vibration. S2, A2, and D1 would all be able to be abnormal due to flaws.

We find that shortcomings commonly happen in two layers:

(1) Sensor layer. False sensor value is a typical issue in the sensor layer. To begin with, physical damage or defect leads sensors to report inaccurate and even wrong sensor values. Likewise, previously unseen circumstances may cause sensors to work beyond their abilities. For example, sensors on spacecraft may come across unexpected conditions.

(2) Control system. CPSs ordinarily hold the dynamic running characteristic, which means there are consistently circumstances that may not be covered during the system design stage. For example, different orders and timings of events in the PLC code can cause object crashes of an assembly line in modern plants.

A. Detection strategies:

Deep Learning methods choose their detection strategies from three aspects:

Input data. Deep Learning methods first need to decide what type of data to take as input, which depends on specific anomalies they tend to detect. Based on the layer and source where data is collected, we conclude four types of input data: (1) Sensor and actuator data. (2) Network traffic data. (3) Systemcalls and logs. (4) Time-series data, which is pre-processed sensor, network, and log data in numeric time-series form. Deep Learning methods adopt semi-supervised and unsupervised learning to resolve the lack of labeled data (especially anomalous data).Neural network design. Deep Learning methods adopt different neural network designs based on input data and tasks. The deep network can be stacked models (e.g., LSTMs) or hybrid combinations of models (e.g., the combination of LSTM and CNN). Although neural network designs can be in various forms, we found several basic models used to build the neural network. (1) RNN: LSTM models (one type of RNN) are often used to capture characteristics of time-series data. (2)Autoencoder: Autoencoders are applied to handle imbalanced data and achieve unsupervised learning.

(3)CNN:CNN models can capture correlations and context information of multivariate measurement data.

Anomaly scores: There exist three metrics to calculate the detection error: (1) Prediction error:Deep Learning methods take past data as input to predict future sensor or actuator values. Then, the error between predicted and real values is measured. Anomalous data usually deviate from predicted values.

(2) Reconstruction error: Input data is fed to the model and compressed to hidden layers, which represents low dimensional space. The data is then reconstructed to the size of the original dimension. Similarly, the error between reconstructed and origin values is calculated. A threshold of error is usually selected to identify anomalous data.

(3) Predicted label or class: If labeled data is relatively sufficient in some domain (e.g., SWaT testbed in ICS), DLAD models can be trained to predict labels of input data. The assumption is that latent features learned from neural networks can be used to identify anomalies. We observe very few methods to adopt this design since a large quantity of labeled data needs profound manual effort.

C.Implementation and evaluation metrics

We summarize the implementation of existing work with an emphasis on platforms where information is collected. Then, metrics that are used to evaluate the effectiveness and performance of Deep Learning methods are identified.

Implementation: As information driven techniques, Deep Learning methods consume a large amount of information to prepare and test models. We summarize three types of environments where information is collected: (1) Data from real-world systems.

(2) Test bed. Researchers construct scaled-down yet entirely utilitarian test beds, where experiments should be possible without the danger of damaging real CPSs.

(3) Simulation. The advantage of information from real-world systems is that it reflects the inherent principle of real systems, although the information is difficult to harvest and the number of systems is limited. Recreation is easy to operate yet cannot capture problems that lone exist in real systems.

A scaled-down test bed could balance the information distortion and operability. Likewise, atypical information can be collected from real-world systems and manually created. There can be insufficient real-world bizarre information since anomalies are difficult to harvest. For example, in brilliant vehicles and medical space, anomalies in real devices may place human lives in danger. So existing studies tackle this problem by manually creating three sorts of anomalies:

(1) Point anomaly. Through investigating anomalies that can happen, several independent abnormal cases can be injected into the normal information series. For instance, Taylor et al. and Russo et al. injected several assault cases into the sequence of CAN transport packets.

(2) Statistical anomaly. Anomalies that follow certain statistical patterns are injected into normal information as an abnormal period.

(3) Simulated assaults. Different assaults are simulated in the testbed, where sensor values and system logs can be easily collected. Zhang et al. created cyber assaults in transactive energy systems.

C.Evaluation metrics:

Metrics are proposed to measure the effectiveness of Deep Learning methods. We conclude that the most commonly used metrics are precision, recall, and F1 score. Given imbalanced datasets, these metrics consider false positives and false negatives, which are better than metrics such as accuracy. The precision is defined as

$$TP/(TP + FP) \qquad (1)$$

Where TP stands for True Positives and FP means False Positives. The recall is defined as

$$TP/(TP + FN) \qquad (2)$$

WhereFN denotes False Negatives. F1 is defined as

$$2*Precision*Recall/(Precision+Recall) \qquad (3)$$

Also, the Receiver Operating Characteristic (ROC) curve is used to manage trade-offs between FP and TP. Meanwhile, methods are often compared with baseline methods to examine the improvement. Some error-based metrics are also applied to measure the prediction and reconstruction performance such as Mean AbsoluteError (MAE) and Relative Errors (ReErr) .

III. UPCOMING TECHNIQUES

1. Applying filters before Deep Learning methods to improve efficiency:

Applying Deep Learning methods in ICS, where running environments are usually resource constrained, should consider the efficiency factor. A lightweight and efficient conventional detecting method could be utilized before Deep

Learning methods to decrease information to be checked significantly. Researchers have proposed a combined anomaly detection framework. The primary idea is to initially apply a Bloom filter to traffic information and then pick dubious packets to the subsequent LSTM-based detector[9,10]. The quick and lightweight filter reduces the burden of the LSTM detector, which enhances the detection efficiency. This method means to identify cyber assaults in the correspondence layer of a SCADA system. The assault types include injecting noxious commands (e.g., state, parameter, and capacity code) and DoS assaults. Additionally, the LSTM detector stacks two LSTM layers using signatures of previous packets to predict the signature of the next packet. Then, the predicted signature is checked to examine whether it is in the normal signature database. The method is evaluated on a gas pipeline system in a laboratory environment, which outperforms baseline methods (e.g., Bayesian Network, Detachment Forest) in the recall, exactness, and F1 score[11].

2.Deep learning-based feature representation: We identify three types of feature representation in Deep Learning methods:

(1)crude information (directly fed to models)

(2)information processing (e.g., inner items of two sensor time series)

(3)deep learning-based embedding. Information processing helps to identify discriminative characteristics of information, which is likewise used in conventional detection methods.

We find that deep learning methods are utilized to integrate features and reduce dimensions of feature space. For example, researchers have proposed deep autoencoders to automatically compress crude contribution to lower-dimension hidden layer representation, which further is utilized as the contribution of the subsequent neural network [12,13]. Despite the two works [14, 15] utilizing the hidden layer to represent features, the real neural network detecting anomalies can be different. One [16] takes sensor value and uses LSTM to generate prediction errors, while the other [16] takes traffic information and uses autoencoder to generate reconstruction errors. The two methods are evaluated on information from testbeds. When expert knowledge is limited (e.g., face a new network protocol), this can be very useful.

3.One sensor or actuator is one-dimension information (e.g., time-series), numerous LSTM-based Deep Learning methods are proposed to learn temporal behaviours of the information. However, there exist correlations among several different sensors and actuators, which reflect logical relations in the control system. In other words, there are interdependent relationships among sensors and actuators. Hence one challenge is to capture context (temporal, spatial, and logical) features in multi-dimensional (time-series of multiple sensors and actuators) information. To this end, CNN can extract features of multi-dimensional information together through convolution operations. Several approaches receive a convolutional layer as the principal layer of the neural network to get correlations of multiple sensors in a sliding time window. Further, the extracted features are fed to subsequent layers to generate yield scores.

These methods can be employed to detect the two assaults and blames. All methods take sensor and actuator value as info and generate prediction error or predicted labels. Meanwhile, other researchers have utilized RNN to take the yield of the CNN layer and form the prediction layer [17,18]. Moreover, the two methods use datasets from real modern plants. Precision, recall, F1, and ROC are evaluation metrics.

4.Exploration of GAN-based methods: The researchers have also proposed a GAN-based framework to capture the spatial-temporal correlation in the multi-dimension information[19]. Both the generator and discriminator are utilized to detect anomalies by reconstruction and segregation errors. Likewise, LSTM models are used to assemble the generator and discriminator. The framework takes sensor and actuator values as information and means to detect false control signals. Compared to a GAN-based anomaly detection method that isn't focused on ICS, this method finds that capturing temporal correlation is the key to improve performance. The method outperforms baseline methods (e.g., Principal component examination, One-Class SVM, K-Nearest Neighbour, Feature Bagging) in precision, recall, and F1. This is an interesting attempt to utilize GAN-based models. Additionally, a well-tuned generator can be used to produce training information.

5.Applying conventional and Deep Learning methods in parallel through ensemble learning. We have introduced that conventional methods can be used as filters before applying Deep Learning methods. However, to increase the precision, these two sorts of methods can be placed parallelly to learn the characteristics of information.

More recently a framework has been proposed called MBPF that ensembles two components:

(1) A statistical method named TBATS (Trigonometric Box-Cox transform, ARMAerrors, Trend, and Seasonal components), and

(2) Multi-branch Deep Network Component. To begin with, seasonality evaluation and outlier elimination are applied to remove noise.

Then, pre-processed information is fed to TBATS and deep learning models simultaneously to capture linear and sequential relations. At last, a Multi-Layer Perceptron (MLP) takes the yield of TBATS and deep learning models, which will vote between the two methods and predict the next value. The MBPF framework can analyse any time-series information. The Mean Absolute Error (MAE) and Root Mean Square (RMSE) are utilized to measure prediction errors. Evaluated on a real-world SCADA water supply system, the method outperforms baseline methods (e.g., Multilayer Perceptron, Stacked LSTM, Regularized LSTM) when measured by MAE, RMSE, Absolute deviation (AbsDev) and Relative Errors (ReErr).

IV. FUTURE DIRECTIONS

Determine the anomaly threshold automatically and adaptively. We argue that the threshold ought to be decided: (1) Automatically:The conventional threshold tuning process isn't efficient and error-prone. To this end, Su et al.utilize the Extreme Value Theory (EVT) to learn the threshold automatically. The key idea is to use a generalized Pareto circulation (GPD) to fit extreme values. Prediction errors of training datasets are used to optimize the threshold. No information conveyance supposition that is needed. Another method is to test a series of threshold values at a fixed interval and check the performance. Intuitively, the value that produces the best result can be selected. (2) Adaptively. A threshold is decided and fixed when a model is trained on a known dataset. However, with the development of CPSs, the limit of anomalies is changing. The threshold ought to evolve as new information comes. A naive strategy is to update the model regularly based on newly collected information. Then, a threshold is generated according to the information. Moreover, online learning could be adopted to let models learn from recent incremental information. Meanwhile, when each time the model is updated, a new threshold is calculated to replace the bygone one. Benchmarks with sufficient labeled and real-world atypical information. To date, we have not discovered numerous benchmarks in CPSs that can be used to compare different Deep Learning methods. Although there exist some frequently used datasets[20] (e.g., SWaT), different Deep Learning methods tend to tailor the dataset and receive the processed information all alone. We suppose that benchmarks in each CPS area (e.g., aerial systems) can help to improve the evaluation process[22]. Different methods may compare performances on the same benchmark. Specifically, we conclude several requirements for benchmarks.

(1) Cover enough information types. Ideally, sensor, actuator, network, and control system logs information can be provided. Deep Learning methods can choose any type of information based on their design goals. Likewise, some models tend to work better on specific information types (e.g., sensor time-series information), which could be produced separately.

(2) Include labeled peculiar information. One challenge to evaluate Deep Learning methods is the absence of labeled anomalies. Researchers have to design and simulate assault or issue cases. Standard and rich assault information and cases can improve detection performance and reduce information processing efforts.

(3) Collect from the real world. Although reproduction is widely adopted in certain spaces (e.g., shrewd grid) due to hardware limitations, real measurements and anomalies can represent the status of systems better. For example, the sequential order and interval of packets in CAN transport traffic in a shrewd vehicle can be utilized as factors to decide whether there is an anomaly.

Recreation may not completely contain and represent these significant factors. Enhance the running performance to a real-time level. We observe that numerous studies in the brilliant vehicle area discussed the running performance of Deep Learning methods. This is because the response time is basic to try not to devastate accidents in shrewd vehicles. To make Deep Learning methods more pragmatic, we argue that running performance is significant in other CPS systems too. Concretely, the design can be improved from two aspects.

(1) Accept real-time input measurements. Instead of using information from offline datasets, Deep Learning methods could acquire online real-time measurements and traffic from have systems. The information sum, sampling rate and format can be decided based on computing resources and network architectures. For example, Deep Learning methods that sudden spike in demand for edge devices can achieve a high detection speed, which is owing to powerful computing capacity.

(2) Take real-time activities. While it is essential to detect anomalies, activities to prevent calamitous losses can likewise be adopted. In some sense, moves ought to be made into account when design and train DLAD models. For example, when designing the misfortune work, we could concentrate how to choose appropriate activities in terms of different anomalies. Locate the peculiar device or root cause.

The detection performance (e.g., true positives, precision) is high in current Deep Learning methods. However, the area and the root cause of the anomaly is usually not identified. Users actually don't have the foggiest idea where an anomaly is from and how to handle the anomaly even Deep Learning methods detect strange status. Moreover, anomalies in different pieces of CPSs present different effects. We argue that Deep Learning methods could improve the detection granularity to component level.Once an anomaly is identified, the compromised device is likewise recognized. Then certain moves could be made to prevent the misfortune. Further, this process can be automatically conducted without the intervention of users. For different CPSs and problems, different compatible neural network architectures can be adopted. We observe that there exist ordinary information types and anomalies in different CPSs. In ICS, sensor time-series measurement information is normally collected. Gradual sensor and sudden actuator change anomalies will break time relations in the information. LSTM-based models and variations are utilized to capture such time relation. Meanwhile, FDI assaults are prevalent in the savvy grid. We find that Deep Learning methods are used to help conventional state estimator methods. LSTM and autoencoder can both be adopted. Moreover, assaults on the CAN transport system in ITS are generally seen. In this way LSTM and CNN are used to capture both time relation and context information (e.g., packet order and content). In aerial systems, most anomalies are injected. LSTM-based methods are utilized to capture time relations. We suggest that researchers' custom their models based on these findings.

V. CONCLUSION

In this work, we systematically reviewed the current research efforts on deep learning-based anomaly detection methods in cyber-physical systems. To this end, we initially propose a scientific classification to recognize the key properties of Deep Learning methods. Further, we highlight prevailing new Deep Learning methods and research findings under the light of our scientific classification. We additionally collect openly available datasets that can be used in Deep Learning methods. To motivate future research in this area, we present our findings, impediments of existing work, and possible future directions to improve Deep Learning methods. Our examination contributes guidance to design down to earth Deep Learning methods and understanding of the current research trend.

REFERENCES

[1] Leo H Chiang, Evan L Russell, and Richard D Braatz. 2000. Fault detection and diagnosis in industrial systems. Springer Science & Business Media.

[2] Kyong-Tak Cho and Kang G Shin. 2016. Fingerprinting electronic control units for vehicle intrusion detection. In 25th USENIX Security Symposium (USENIX Security 16).911–927.

[3] Safety Pilot Model Deployment Data. 2020. Retrieved Jan 07, 2020 from https://catalog.data.gov/dataset/safety-pilotmodel-deployment-data

[4] REFIT Smart Home dataset. 2017. Retrieved Dec 12, 2019 from https://repository.lboro.ac.uk/articles/REFIT_Smart_Home_dataset/2070091.

[5] Alysha M De Livera, Rob J Hyndman, and Ralph D Snyder. 2011. Forecasting time series with complex seasonalpatterns using exponential smoothing. J. Amer. Statist. Assoc.106, 496 (2011), 1513–1527.

[6] Qingyu Deng and Jian Sun. 2018.False Data Injection Attack Detection in a Power Grid Using RNN.In IECON2018-44th Annual Conference of the IEEE Industrial Electronics Society.IEEE, 5983–5988.

[7] BenediktEiteneuer, NemanjaHranisavljevic, and Oliver Niggemann. 2019. Dimensionality Reduction and Anomaly Detection for CPPS Data using Autoencoder. (Feb. 2019), 1286-1292. https://doi.org/10.1109/ICIT.2019.8755116 ISSN:2641-0184.

[8] MellitusEzeme, AkramulAzim, and Qusay H Mahmoud. 2017. An imputation-based augmented anomaly detectionfrom large traces of operating system events. In Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies.43–52.

[9] Mellitus O Ezeme, Qusay H Mahmoud, and AkramulAzim. 2018. Hierarchical attention-based anomaly detection model for embedded operating systems. In 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). IEEE, 225–231.

[10] Okwudili M Ezeme, Michael Lescisin, Qusay H Mahmoud, and AkramulAzim. 2019. DeepAnom: An Ensemble Deep Framework for Anomaly Detection in System Processes. In Canadian Conference on Artificial Intelligence. Springer, 549–555.

[11] Okwudili M Ezeme, Qusay H Mahmoud, and AkramulAzim. 2019. Dream: deep recursive attentive model for anomaly detection in kernel events. IEEE Access 7 (2019), 18860–18870.

[12] Cheng Fan, Fu Xiao, Yang Zhao, and Jiayuan Wang. 2018. Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. Applied energy 211 (2018), 1123–1135.

[13] ChengFeng, Tingting Li, and DeephChana. 2017. Multi-level anomaly detection in industrial control systems via package signatures and lstm networks. In 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 261–272.

[14] P Ferrari, S Rinaldi, E Sisinni, F Colombo, F Ghelfi, D Maffei, and M Malara. 2019. Performance evaluation of full-cloud and edge-cloud architectures for Industrial IoT anomaly detection based on deep learning. In 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT). IEEE, 420–425.

[15] FireEye. 2020. A View Into The Top 20 Cyber Attacks on ICS Networks | FireEye. Retrieved Feb 18, 2020 from https: //www.fireeye.com/solutions/industrial-systems-and-critical-infrastructure-security/wp-top-20-cyberattacks.html

[16] Flightradar24. [n.d.]. Live Flight Tracker - Real-Time Flight Tracker Map. https://www.flightradar24.com/

[17] JairoGiraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. 2018. A survey of physics-based attack detection in cyber-physical systems.ACM Computing Surveys (CSUR) 51, 4 (2018), 1–36.

[18] Jonathan Goh, Sridhar Adepu, KhurumNazirJunejo, and AdityaMathur. 2016. A dataset to support research in the design of secure water treatment systems. In International Conference on Critical Information Infrastructures Security.Springer, 88–99.

[19] Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. 2017. Anomaly detection in cyber physical systems using recurrent neural networks. In 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE). IEEE, 140–145.

[20] Lachlan Gunn, Peter Smet, Edward Arbon, and Mark D McDonnell. 2018. Anomaly Detection in Satellite Communications Systems using LSTM Networks. In 2018 Military Communications and Information Systems Conference (MilCIS). IEEE, 1–6.

[21] EdanHabler and AsafShabtai. 2018. Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages. Computers & Security 78 (2018), 155–173.

[22] Hacking and Countermeasure Research Lab. 2020. CAN-intrusion-dataset (OTIDS). Retrieved Jan 07, 2020 from http://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset.